SONDERHEFT 24 ÖS 100,-/Sfr. 14,- Lit. 12000/hfl. 18,-/dkr. 72,- DM 14,-

Markt&Technik



Tips & Tricks die jeder braucht

- ★ Die besten PEEKs und POKEs gesammelt und sortiert
- ★ Auto-Save: kein Programmverlust bei Stromausfall

Das Ende aller Tippfehler

- ★ Automatische Textkorrektur für Master-Text und Vizawrite

Utilities mit Pfiff

- ★ Der Super-Packer: Software optimal gekürzt
- ★ Grafik: Bilder einblenden wie im Film
- ★ Drucker: Hardcopys für viele Drucker

Basic-Compiler der Spitzenklasse zum Abtippen

Alle Programme auch auf Diskette erhältlich

RAUSHOLEN, WAS DRINSTECKT. MIT DER ORIGINAL-PERIPHERIE FÜR DEN COMMODORE 64.



Disk Drive 1581
Zugreifen – blitzschnell!
Mit 3.5", Speicherkapazität 800 KB.



BTX-Decoder
Jetzt werden andere Seiten aufgeschlagen. Mehr als 600.000 BTX-Seiten abrufbereit. Auch für Commodore 128 (D) einsetzbar.



1802 Farbmonitor
Der sitzt auf dem Punkt!
13" Bildschirm, PAL-Fernsehnorm.



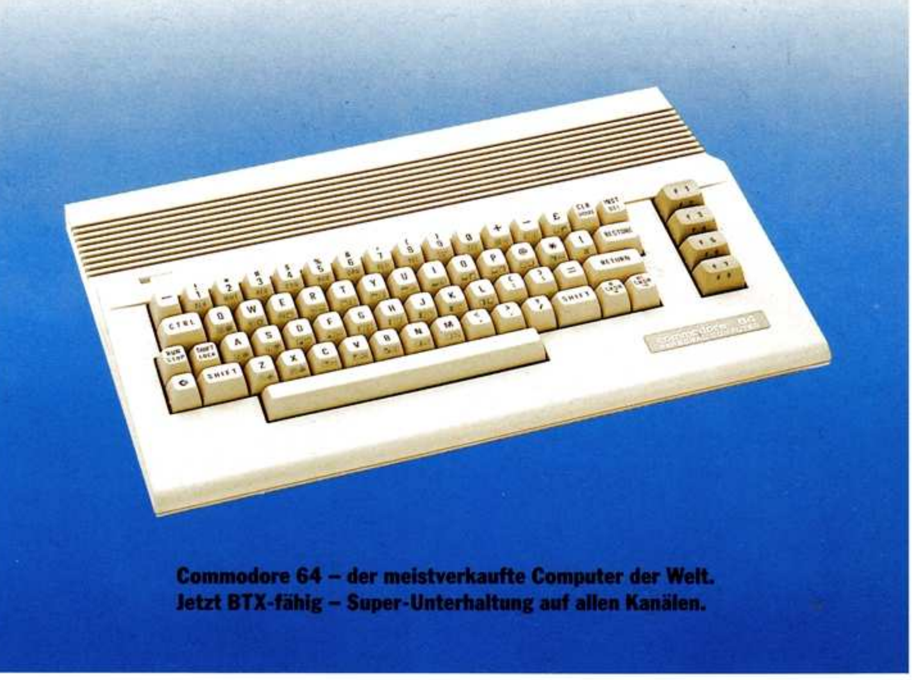
Disk Drive 1541
Die steckt 'ne ganze Menge weg.
Mit 5.25", Speicherkapazität 170 KB.



RAM-Erweiterung 1764
Darf's ein bißchen mehr sein?
Zusätzlicher Speicher 256 KB, bei Verwendung als RAM-Disk Zugriffszeit bis zu 1 MB/Sek.



MPS 1200
Damit alle den richtigen Eindruck bekommen!
120 Zeichen/Sek. Schönschrift und Normalschrift, internationale Zeichen, Sonderzeichen, Grafikzeichen.

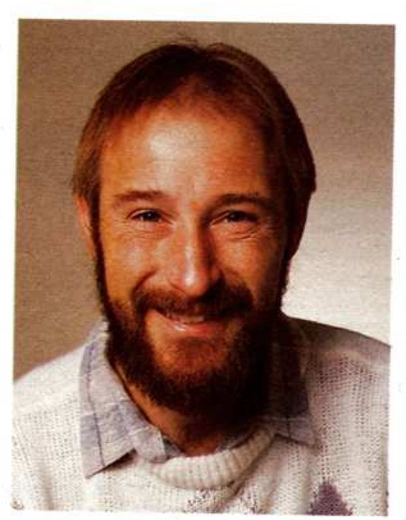


Commodore 64 – der meistverkaufte Computer der Welt.
Jetzt BTX-fähig – Super-Unterhaltung auf allen Kanälen.

Computer sind wie Anzüge.
Es muß einfach alles passen, damit man sich wohl fühlt.
Original Commodore-Geräte sind optimal auf Ihren 64er abgestimmt.
Da paßt alles – sofort!
Commodore – Marktführer bei Mikrocomputern.



Das Käfer-Syndrom



Wer kennt es nicht, dieses rundliche Auto für jedermann. Man taufte es schlicht »Käfer« und es lief und lief und lief - ähnlich wie unser C64. Was waren die Gründe für den Erfolg dieses Autos? Das perfekteste Fahrzeug war es ja nicht. Aber man konnte nahezu alles preiswert bekommen. Und den Käfer-Freaks fiel immer wieder etwas Neues ein. Von der Selbstbauvariante für den Strand, dem Buggy, bis zum spurverbreiterten Superkäfer mit Porsche-Motor war alles dabei.

Betrachten wir einmal den C64 im Vergleich. Als den perfektesten Computer kann man auch ihn nicht bezeichnen. Aber das ist es vielleicht gerade, was ihn so beliebt und interessant macht. Etlichen fällt ein Tip oder Trick ein, um diesen Computer besser zu beherrschen. Immer wieder entwickeln C64-Fans eine neue Softwarehilfe, die den Umgang mit dem Computer vereinfacht und eine Bereicherung darstellt. Wir wollen Ihnen diese Tips, Tricks und Tools, die jedem C64-Besitzer das Leben erleichtern, mit diesem Sonderheft zur Verfügung stellen.

Grafiken können Sie jetzt einblenden wie im Film. In Dia-Shows lassen sich Bilder weich überblenden. Die Basic-Erweiterung »Alan« macht das Erstellen von interessanten Grafiken zum Kinderspiel und verfügt über einen erheblich verbesserten Basic-Editor, mit dem Sie sogar Listings aufwärts und abwärts scrollen können.

Textverarbeitung ist eine der verbreitetsten Anwendungen. Trotz Computer sind Tippfehler noch an der Tagesordnung. Das können wir ändern für Master-Text und Vizawrite. Deutsche Spell-Checker überprüfen die Rechtschreibung, und der Wortschatz läßt sich schnell ergänzen. Für beide Spell-Checker kann man übrigens die gleiche Wörterdatei benutzen.

In Basic lassen sich zwar recht einfach Programme entwickeln, aber oft stößt man an Grenzen. Besonders, wenn es um zeitkritische Abläufe geht, ist häufig »guter Rat teuer«. Zwei Artikel warten auf Sie. Wir zeigen Ihnen, wie Sie auch in Basic erstaunlich schnelle Programme erzeugen können, und bieten Ihnen einen Basic-Compiler zum Abtippen, der den Vergleich mit kommerziellen Produkten nicht zu scheuen braucht. Jeder Programmierer kennt es. Bei der Entwicklung eines Programms fällt einem immer wieder etwas Neues ein. Nach und nach legt die Software enorm an Umfang zu. Jetzt kommt die radikale Abmagerungskur. Der »Super-Packer« stutzt Ihre Software radikal - egal ob es Programme oder Grafiken sind, alles wird kleingemacht und läßt sich in neuer Kürze wieder auf der Diskette unterbringen. Zudem werden Sie mit den Grundlagen des Packens und Komprimierens vertraut gemacht. Zwei Kurse bieten alle wichtigen Informationen, für Einsteiger und Profis. Fundamentierte Grundlagen sind in der Computerei überhaupt ein wesentlicher Punkt.

Eine interessante Hilfe für Basic-Programmierer ist, Maschinensprache-Routinen des Betriebssystems direkt zu nutzen. Basic-Programme werden dadurch wesentlich schneller und kürzer.

Auch der Umgang mit der RS232-Schnittstelle ist vielen ein Buch mit sieben Siegeln. Wir knacken diese und bringen Licht in diese Angelegenheit. Interessant ist dies besonders, da wir Ihnen ein Programm vorstellen, das mit einer Übertragungsrate von 4800 Bit/s arbeiten kann.

Natürlich gehören zu einem »Tips & Tricks«-Sonderheft auch viele dieser kurzen Hilfen. Wir haben die Tips & Tricks ebenso wie PEEKs & POKEs für Sie gesammelt und übersichtlich geordnet. Egal, ob es sich um Grafik, Scrolling, Cursorsteuerung, Floppy, Programmierschutz oder andere Effekte handelt, nach Themenbereichen alphabetisch sortiert, läßt sich der richtige Trick schnell finden.

Etliche Neuigkeiten gibt es ebenfalls aus der Trickkiste. Vielleicht haben Sie sich auch schon einmal kräftig geärgert, als Sie stundenlang in die Programmierung einer neuen Software vertieft waren — und plötzlich fiel der Strom aus. »Timesave« rückt diesem Problem zu Leibe und speichert in regelmäßigen Abständen automatisch. Das ist aber längst nicht alles. Lassen Sie sich überraschen.

Gottfried Knechtel

Ihr Gottfried Knechtel
(Leitender Redakteur)

INHALT

Tools: Grafik

Tools: Drucker

Tools: Textverarbeitung

Tools: Floppy

Tools: DFÜ

Tools gemischt

Grundlagen

Tips & Tricks

Tabellen

Eingabehilfen

Sonstiges

Alle Programme aus Artikeln mit dem ■-Symbol finden Sie auch auf der Programmservice-Diskette zu diesem Sonderheft

Einblenden wie im Film

Hires-Bildschirme weich überblenden oder Sprites punktweise auf dem Spielfeld erscheinen lassen, machen eine einfache Verschieberoutine zu einem kompletten Grafik-Utility. Hinter »Super-Dump« verbirgt sich eine komfortable Routine für tolle Effekte.

Die Routine (Listing 1) wird mit
LOAD"SUPER-DUMP,OBJ",8,1
(,1,1 für Datasette) geladen. Der Aufruf kann sowohl von Basic, als auch von Assembler aus erfolgen. Zunächst aber die Syntax für Basic:

SYS 49152, Quellblock, Länge, Zielblock, Muster, Tempo, Art

Die Parameter haben dabei folgende Bedeutung:

Quellblock

Anfangsblock des Quellbereichs, der kopiert werden soll. Man erhält ihn, wenn man die Anfangsadresse durch 256 teilt, zum Beispiel 8192 durch 256 (Hires-Grafik) ergibt die Blocknummer 32.

Zulässige Werte: 0 bis 255.

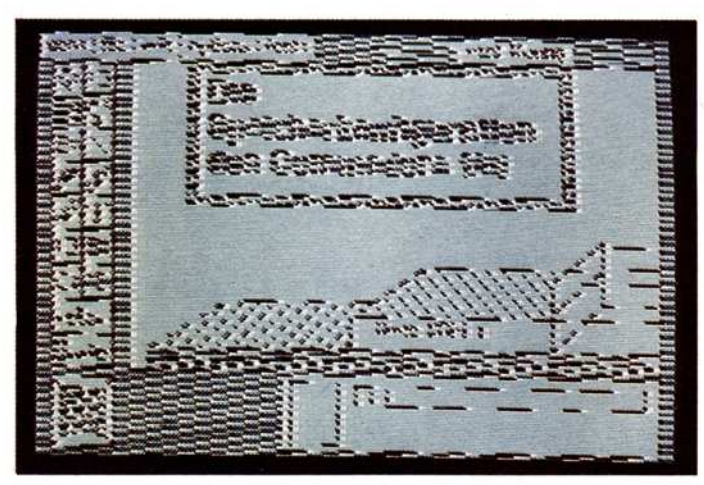


Bild 2. . . . und kurz vor dem Ende der Bilderzeugung

Länge:

Wird ebenfalls in Blöcken angegeben.

Zulässige Werte: 0 bis 255.

Zielblock:

Anfangsblock des Bereichs, in den hineinkopiert werden soll.

Zulässig sind Werte von 4 bis 255 (0 bis 4: Zeropage).

Muster:

Bestimmt das effektvolle Kopierverfahren, auf das noch genauer eingegangen wird.

Möglich sind Werte von 0 bis 255, wobei allerdings bei Werten größer als 127 der Wert 127 gilt.

Tempo:

Hier kann man durch Werte von 0 bis 255 bestimmen, wie schnell die Routine abläuft.

Art:

Setzt man 0 ein, wird byte-weise kopiert, bei allen anderen Werten wird bit-weise kopiert.

Alle benötigten Parameter sind von Basic aus durch geringen Rechenaufwand zu erhalten und zu übergeben. Mögliche Fehleingaben führen nicht zum Systemabsturz, sondern zur korrekten Fehlermeldung, wenn etwa Werte

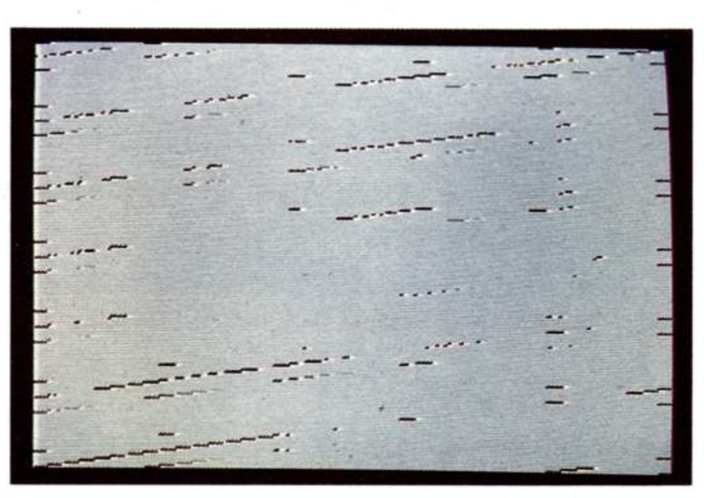


Bild 1. Die Bilder 1 bis 3 zeigen verschiedene Phasen der Einblendung. Am Anfang . . .

größer als 255 eingegeben werden oder die Länge des zu kopierenden Bereichs sämtliche Speichergrenzen sprengt.

Tips für Assembler-Programmierer

Für Benutzer der Maschinensprache verkürzt sich das Programm um den Teil, der die Parameter aus dem Basic-Text einliest. Da dieser Teil auch die Prüfung auf Zulässigkeit enthält, dürfen nur korrekte Werte übergeben werden, bevor die Hauptroutine angesprungen wird. Im folgenden die Speicherstellen, in die die Parameter geschrieben werden müssen:

$02a7 – Anfangsblock des Quellbereichs
$02a8 – Länge des Bereichs
$02a9 – Anfangsblock des Zielbereichs
$02aa – Parameter für Kopiermuster
$02ab – Geschwindigkeit
$02ac – Byte- oder Bitweise

Sind diese Parameter korrekt übergeben, kann die Hauptroutine durch »JSR $c070« aufgerufen werden. Benötigt wird allerdings der Bereich von $c065 bis c1a0!

Beschreibung der Funktionen:

Nachdem die Bedienung geklärt ist, nun zu den Funktionen und Fähigkeiten von Super-Dump.

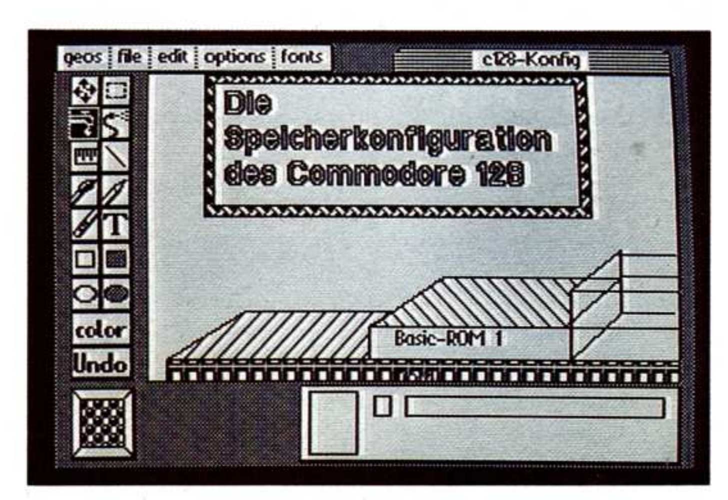


Bild 3. Das Bild ist fertig eingeblendet

Es werden schnell viele Anwendungen deutlich: Ein-, Aus- und Überblenden von Hires-Grafik, Bildschirm-Masken oder Sprite-Grafik. So kann man zum Beispiel Spielfiguren effektvoll »zerbröseln« oder ins Spielfeld »beamen«! In Bild 1 bis 3 sehen Sie drei verschiedene Stadien eines solchen Überblendvorgangs.

Ferner kann man Super-Dump als schnelle Kopierroutine benutzen, um zum Beispiel durch

```
SYS 49152, 160, 32, 160, 0, 0, 0
```

das Basic-ROM ins RAM zu kopieren und dort zu verändern. Durch

```
POKE56334,0:POKE1,51
SYS 49152,208,16,xxx,0,0,0
POKE1,55:POKE56334,1
```

wird der Zeichensatz aus dem Character-ROM ab xxx ins RAM kopiert. Bei solchen Anwendungen sollte man natürlich byteweise kopieren und als Geschwindigkeit 0 wählen.

Eine weitere Möglichkeit besteht darin, durch Super-Dump größere Speicherbereiche als den Hires-Grafikspeicher zu löschen oder mit einem bestimmten Byte zu füllen. Man löscht einen Block und kopiert ihn beliebig oft in einer Schleife in den Bereich, den man löschen möchte!

In der Erklärung der Parameter wurde die Kopiermusterzahl erwähnt, die das Kopierverfahren beeinflußt. Diese Zahl von 0 bis 127 wird zuerst verdoppelt und dann um 1 erhöht. So erhält man eine ungerade Zahl von 1 bis 255. (Wichtig für Assembler-Programmierer: Nur ungerade Werte in $02aa übergeben, da die Routine sonst nicht korrekt arbeitet!) Nachdem die Routine diese Zahl erhalten hat, wird ein Zähler angelegt, der 256mal um diese ungerade Zahl erhöht wird. Wird der Zähler größer als 255, subtrahiert der Prozessor automatisch 256, so daß man alle Zahlen von 0 bis 255 in scheinbar wahlloser Reihenfolge erhält. Mit diesen Zahlen werden alle Bytes eines Blocks kopiert, das heißt nicht nacheinander, sondern scheinbar wahllos.

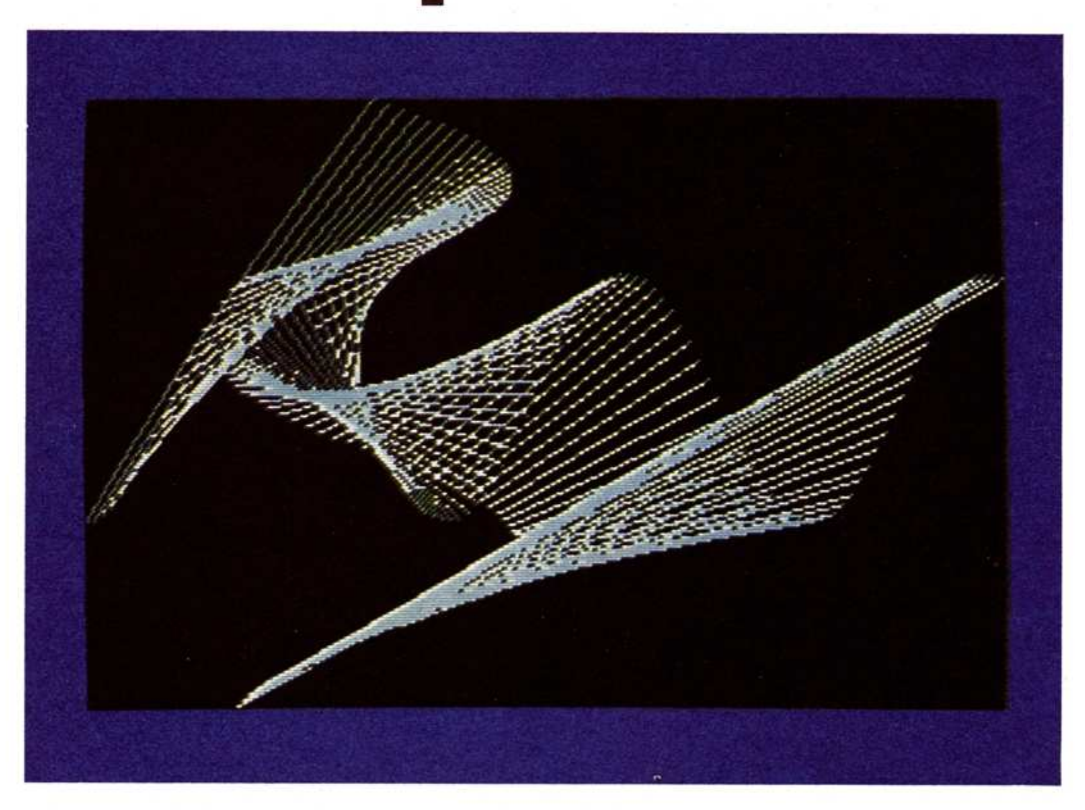
Eine Diashow läßt sich somit leicht erstellen, indem die Bilder in einen unsichtbaren Bereich geladen und zum Beispiel auf Tastendruck in den sichtbaren Bereich kopiert werden.

(Holger Schemel/sk)

```
Name : super-dump.obj    c000 c10f
-----------------------------------
c000 : 20 fd ae 20 9e b7 8e a7  00
c008 : 02 20 fd ae 20 9e b7 8a  5a
c010 : 6d a7 02 cd a7 02 b0 01  da
c018 : 60 8e a8 02 20 fd ae 20  17
c020 : 9e b7 e0 04 b0 01 60 8a  fc
c028 : 6d a8 02 cd a8 02 b0 01  83
c030 : 60 8e a9 02 20 fd ae 20  6f
c038 : 9e b7 e0 80 90 02 a2 7f  9d
c040 : 8e aa 02 18 ea ea ea 8a  6e
c048 : 6d aa 02 69 01 8d aa 02  e3
c050 : 20 fd ae 20 9e b7 8e ab  58
c058 : 02 20 fd ae 20 9e b7 8e  b2
c060 : ac 02 4c 70 c0 a2 01 ca  e9
c068 : d0 fd c6 02 d0 f7 60 ea  4d
c070 : ae a8 02 d0 01 60 a9 08  d7
c078 : 8d af 02 a9 01 8d ad 02  ca
c080 : a9 fe 8d ae 02 a0 00 a9  5a
c088 : 00 85 fb 85 fd ad a7 02  ea
c090 : 85 fc ad a9 02 85 fe ae  d9
c098 : a8 02 ad ac 02 d0 07 b1  68
c0a0 : fb 91 fd 4c b6 c0 b1 fd  a1
c0a8 : 2d ae 02 91 fd b1 fb 2d  97
c0b0 : ad 02 11 fd 91 fd ea e6  e5
c0b8 : fc e6 fe ca 8a 48 ad ab  39
c0c0 : 02 85 02 f0 03 20 65 c0  6c
c0c8 : 68 aa d0 ce ad ad 02 18  14
c0d0 : 2a 2a 2a 2a b0 fa 8d ad  54
c0d8 : 02 49 ff 8d ae 02 98 6d  69
c0e0 : aa 02 a8 d0 a8 ad ac 02  7e
c0e8 : f0 1f ce af 02 f0 1a ae  7f
c0f0 : af 02 ad ad 02 18 2a 2a  9f
c0f8 : ea b0 fb ca d0 f8 8d ad  f9
c100 : 02 49 ff 8d ae 02 4c 8d  a0
c108 : c0 60 00 00 00 00 00 00  f9
```

Listing 1. »SUPER-DUMP.OBJ«, eine einfache Routine für tolle Grafikeffekte

Grafik spielend leicht



Alan: eine komfortable Grafik-Erweiterung mit enormer Geschwindigkeit. Kurze Programme erzeugen die schönsten Grafiken. Zusätzlich wird die Programmierung von Windows, Scrolling und ein verbesserter Basic-Editor geboten.

Die Basic-Erweiterung »Alan« stellt viele neue Grafikbefehle zur Verfügung, ohne den Basic-Speicher wesentlich einzuschränken. Immerhin verbleiben Ihnen noch 34 KByte für die Basic-Programmierung.

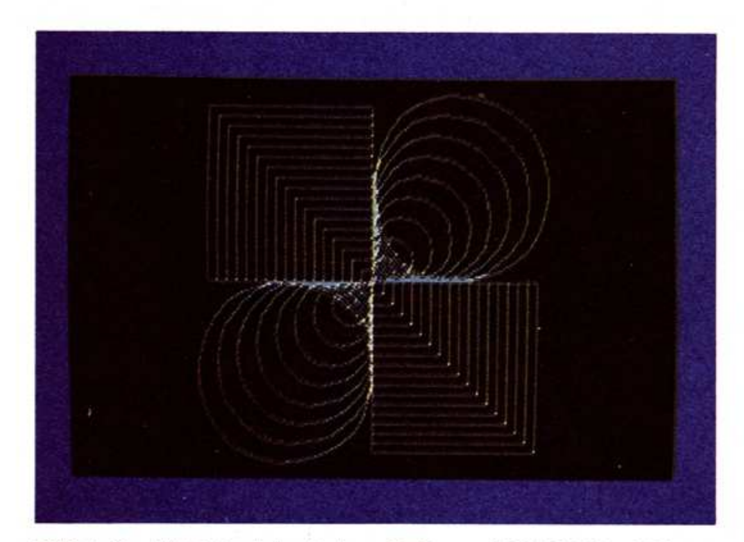


Bild 1. Grafik-Beispiel mit dem REC- und CIRCLE-Befehl

Neben den neuen Grafikbefehlen stehen Ihnen auch noch Editier-, Disketten-, Cursor-, Zeichensatz-, Scroll- und Windowbefehle zur Verfügung. Außerdem bietet »Alan« eine erstaunliche Geschwindigkeit. In den Bildern 1 bis 8 zeigen Ihnen einige Beispiele, was Alan zu bieten hat.

Geben Sie »Lader Alan V7.2« (Listing 1) bitte mit dem Checksummer (Seite 159) ein. »Alan V7.2« (Listing 2) und »Alan II« (Listing 3) müssen mit dem MSE (Seite 159) eingegeben werden. Laden Sie »Lader Alan V7.2« und starten Sie das Programm mit RUN. »Alan V7.2« und die anderen Unterprogramme werden selbständig nachgeladen und aktiviert. Wenn das Einschaltbild und der Cursor erscheint, können Sie die neuen Befehle benutzen. »Alan« hat einen Reset-Schutz, das heißt, nach einem Druck auf einen Reset-Knopf wird »Alan« wieder gestartet.

Listing 4 ist ein Demoprogramm zu Alan. Es erzeugt die Bilder 1 bis 5 (und ein zusätzliches) in erstaunlicher Geschwindigkeit. Lassen Sie sich faszinieren davon. Auf der Programmservice-Diskette werden Sie noch weitere schöne Demos finden.

Diskettenbefehle

CAT

CAT zeigt das Directory der eingelegten Diskette des Laufwerks Nummer 8 an. Durch Drücken von ⟨SPACE⟩ wird der List-Vorgang unterbrochen, danach wieder fortgesetzt. ⟨RUN/STOP⟩ beendet die Ausgaben. CAT führt zu keinem Programmverlust und kann selbstverständlich in eigene Programme eingebaut werden.

STATUS

STATUS liest den Diskettenstatus ein und gibt ihn auf dem Bildschirm aus.

DISK "Befehl"

Man sendet den entsprechenden Disketten-"Befehl" zum Floppy-Laufwerk. Dadurch erspart man sich die Anweisung »OPEN 1,8,15,"Befehl" : CLOSE 1«. Der Diskettenbefehl muß in Anführungszeichen eingeschlossen sein.

BLOAD "Filename"
Dieser Befehl lädt ein Programm von Diskette nach, ohne irgendwelche Zeiger zu verändern. So kann man beispielsweise ein Maschinenprogramm nachladen, ohne daß das Basic-Programm wieder gestartet wird.

Editorbefehle

HELP
Alle neuen Befehle werden am Bildschirm angezeigt.

OLD
Ein mittels »NEW« gelöschtes Programm kann durch OLD wieder zurückgeholt werden. Nach einem erweiterten Reset bei modifizierten Betriebssystemen wird »Alan« mit SYS 49152 reinitialisiert; man kann dann mit OLD das alte Programm wieder zurückholen.

AUTO Anfangswert, Schrittweite
Der AUTO-Befehl gibt die Zeilennummern automatisch aus. Die automatische Ausgabe wird durch Drücken von ⟨RETURN⟩ direkt nach der Zeilennummer aufgehoben.

REN Anfangswert, Schrittweite
Der RENUMBER-Befehl gibt dem gesamten Listing neue Zeilennummern. Nach ⟨RETURN⟩ wird dies ausgeführt. Zu beachten ist, daß Sprungbefehle nicht mit einer neuen Zieladresse versehen werden. Mit dem FIND-Befehl lassen sich die Sprungbefehle aber leicht finden.

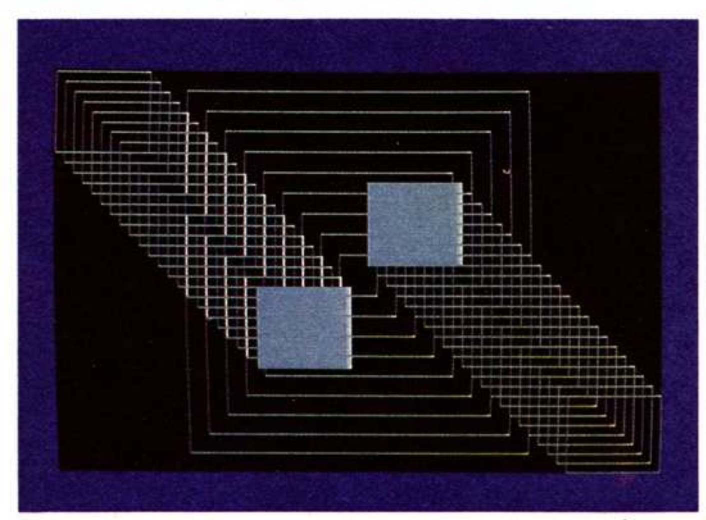


Bild 2. Eine Kombination des REC- und BLOCK-Befehls

DEL Anfangszeile — Endzeile
Es werden die Basic-Zeilen von der Anfangs- bis zur Endzeile gelöscht. Zwischen beiden Parametern muß ein »–« stehen. Ferner müssen die angegebenen Zeilennummern tatsächlich existieren.

LLIST Anfangszeile — Endzeile
LLIST gibt das Listing auf dem Drucker mit Geräteadresse 4 aus. Mit der gleichen Syntax wie beim DEL-Befehl können Listing-Teile gedruckt werden. Läßt man die Zeilennummern weg, wird das gesamte Listing ausgegeben.

LSCROLL A
Mit LSCROLL kann ein Listing gescrollt werden. ⟨F1⟩ scrollt nach oben, ⟨F3⟩ nach unten. Gibt man für die Variable »A« den Wert »1« ein, wird der Befehl aktiviert, bei »0« wieder deaktiviert. Mit ⟨F5⟩ kann der Cursor bei aktiviertem Befehl in die untere linke Ecke positioniert werden.

FIND Ausdruck
FIND durchsucht das Basic-Listing nach dem angegebenen Ausdruck und listet diese Zeile auf. Die so ausgegebenen Zeilen können Sie dann sofort bearbeiten. Der Ausdruck muß direkt hinter dem Befehl stehen; es darf kein ⟨SPACE⟩ dazwischen sein. »FINDSIN(X)« sucht überall nach dem Ausdruck »SIN(X)« im Basic-Listing. »FIND SIN(X)« dagegen sucht im Programm den Ausdruck

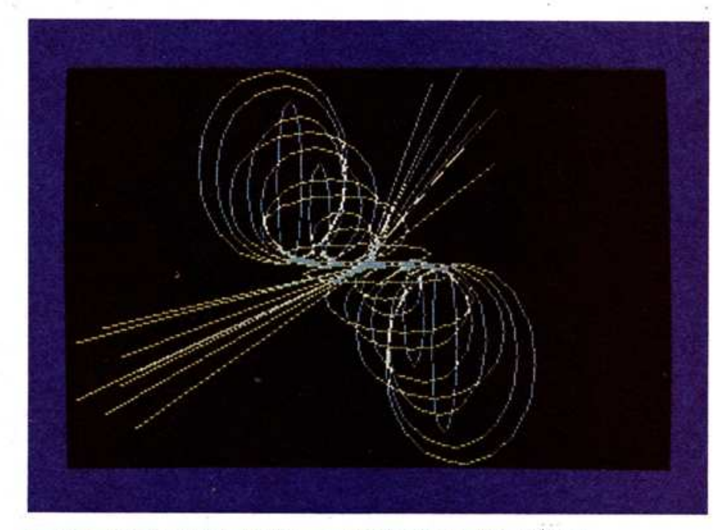


Bild 3. Die Befehle LINE und CIRCLE in Varianten

»⟨SPACE⟩ SIN(X)«. Geben Sie also bei der Benutzung dieses Befehls auf Leerzeichen besonders acht. Die Ausgabe der gefundenen Zeile wird durch ⟨RUN/STOP⟩, nicht aber durch ⟨RUN/STOP RESTORE⟩, unterbrochen.

CEN
Es wird eine am User-Port simulierte Centronics-Schnittstelle aktiviert.

DMODE
Wenn DMODE aktiviert wurde, werden alle Basic-Zeilen gelöscht, die man mit dem Cursor »überfährt«.

DOFF
Durch diesen Befehl wird DMODE deaktiviert.

Cursorbefehle

HOME A
A nimmt den Wert 0 oder 1 an. Bei HOME 0 wird der Bildschirm gelöscht und der Cursor nach 0/0 (linke obere Ecke) gesetzt. HOME 1 löscht den Bildschirm, setzt aber den Cursor auf 0/22. Bei dieser Position beginnt im gesplitteten Bildschirm (siehe Grafikmodus 1) das Textwindow; so kann man mittels »PRINT« direkt in das Text-Window schreiben.

VTAB A
Mit diesem Befehl wird der Cursor auf die A-te Zeile positioniert. Für A sind nur Werte zwischen 0 und 24 erlaubt. Beispiel: VTAB 5 setzt den Cursor in die 5. Zeile.

Zeichensatzbefehle

CCOPY
CCOPY kopiert den Zeichensatz von $D000 nach $3000. Dort kann er dann nach Belieben bearbeitet werden.

CNEU
Es wird der Zeichensatz ab $3000 eingeschaltet.

CALT
Der Zeichensatz ab $3000 wird abgeschaltet und der original Commodore-Zeichensatz wird aktiviert.

Scroll- und Window-Befehle

WGET D$
WGET liest einen String A$ ein. Das geschieht eigentlich genauso wie bei dem Befehl INPUT. Der Unterschied ist

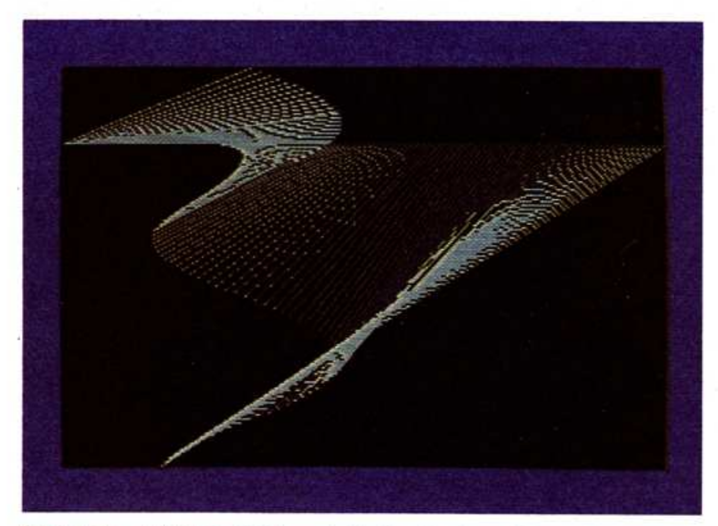


Bild 4. Der LINE-Befehl verknüpft mit einer Sinus-Funktion

der, daß man Sonderzeichen wie »,;:« oder ähnliches eingeben kann. Der eingegebene String wird im Kassettenpuffer ab Adresse 828 zwischengespeichert. So kann dieser String von dem SCROLL Befehl weiterbenutzt werden. Jedoch muß der String, den man einliest, immer wieder neu geschrieben werden, da die Get-Routine des Betriebssystems benutzt wird.

TRANSFER A$
Mit TRANSFER A$ wird der String A$ direkt in den Kassettenpuffer gespeichert. Ein eventuell nachfolgender SCROLL-Befehl kann diesen String nun weiter verwenden. Dies ist dann sinnvoll, wenn man eine Laufschrift erzeugen will, ohne daß der Text von der Tastatur eingelesen wird.

SCROLL A
Der Wert A gibt die Anzahl der Zeichen an, die am unteren Bildschirmrand gescrollt werden. Er kann Werte von 1

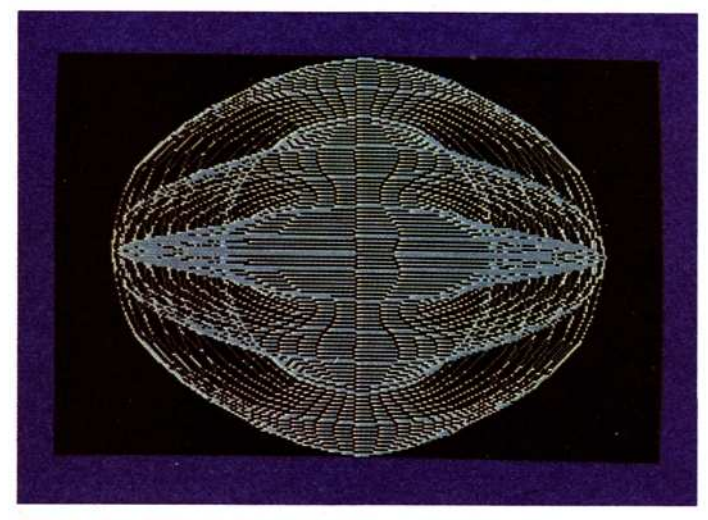


Bild 5. Der CIRCLE-Befehl in Sinus-Kombination

bis 255 annehmen. Der Text steht dabei im Kassettenpuffer, wo er mittels TRANSFER oder WGET abgelegt wurde. Es empfiehlt sich, am Anfang der Zeichenkette zwei Leerzeichen einzugeben. Das Scrollen wird durch den Befehl TEXT beendet.

WINON X, Y, Länge, Breite, Nummer
Mit WINON wird ein beliebiges Window definiert. X und Y geben die Anfangskoordinaten (linke obere Ecke) an. Länge und Breite bestimmen die Größe des Windows. Die Nummer gibt an, welches Window eingeschalten wurde. Es können nur zwei Windows definiert werden, da sie sehr viel Speicherplatz benötigen. Nach der Windowdefinition wird der definierte Bereich vorerst mit Shift-Space aufgefüllt. Die Farbe legt man mittels »POKE 39169,Farbe« fest. Der Cursor wird im Window positioniert und durch eine programminterne Abfrage daran gehindert, es wieder zu verlassen.

WINOFF
WINOFF schaltet das zuvor definierte Window wieder aus. Deswegen werden die »alten« Werte zurückgeholt. Außerdem schaltet das Programm den Interrupt wieder auf normale Konfiguration, und der Cursor ist wieder frei zu bewegen.

COL Rahmen-, Hintergrund-, Zeichenfarbe
Mit COL wird die Rahmen-, Hintergrund- und Zeichenfarbe gesetzt.

Grafikbefehle

HGR A
A nimmt den Wert 0 oder 1 an. HGR 1 schaltet die Grafik auf den ganzen Bildschirm an. Man hat nun eine Auflösung von 320 x 200 Punkten zur Verfügung. Die Bitmap erstreckt sich von $6000 bis $7F40.

Bei HGR 0 wird der obere Bereich des Bildschirms als Grafikfenster verwendet; die unteren fünf Zeilen sind für Text gedacht. Mit HOME 1 wird der Textbildschirm gelöscht und der Cursor nach 0/20 positioniert. Diese Betriebsart kann man gut in Adventures nutzen (oben Grafik, unten Text). Auch diese Grafikbitmap beginnt bei $6000; man hat aber nur eine Auflösung von 320 x 168 Punkten zur Verfügung. Bei HGR 0 können weder Windows definiert werden noch die Laufschrift (SCROLL) eingeschalten werden, da alle drei Routinen den Interrupt verwenden.

TEXT
Mit TEXT wird die Grafik ausgeschaltet und der Textbildschirm wieder aktiviert. Außerdem stoppt dieser Befehl die Laufschrift.

GCLR
GCLR löscht den Grafikbildschirm.

SET Zeichen-, Hintergrundfarbe
Mit SET werden die Farben der Grafik festgelegt.

MODE A
MODE stellt den Zeichenmodus ein, wobei A den Wert 0 oder 1 haben kann. MODE 0 gibt an, daß alle Grafikpunkte gesetzt werden. MODE 1 bewirkt genau das Gegenteil, alle Grafikpunkte werden gelöscht.

INV
Der ganze Grafikbildschirm wird invertiert, das heißt, ein gesetzter Punkt wird gelöscht und ein gelöschter Punkt wird gesetzt.

PLOT X, Y
PLOT setzt Punkte an der X- und Y-Position des Grafikbildschirms. Die X-Koordinate darf dabei den Wert 319, die Y-Koordinate den Wert 199 nicht überschreiten, da sonst eine Fehlermeldung ausgegeben wird.



Der KAUFHOF bringt's zum Ausdruck



SP 120VC Matrixdrucker

- anschlußfertig für alle Commodore-Heimcomputer
- 13 Schriftarten in EDV- und Korrespondenzqualität
- max. 120 Zeichen/Sekunde
- grafikfähig

nur **399,–**



»elite« SP 120VC Matrixdrucker

Dieser Drucker zum Superpreis macht die Gestaltung Ihrer Texte einfach: Sperrschrift, Fettdruck und Schattenschrift stellt er Ihnen zur Verfügung. Hoch- und Tiefstellen sowie Unterstreichen sind natürlich ebenfalls möglich. Dabei druckt er wahlweise in Elite, Pica oder Schmalschrift. Normal, kursiv oder negativ. Im Schnelldruck schafft er bis zu 120 Zeichen/Sekunde, in Korrespondenzqualität bis zu 22 Zeichen/Sekunde. Auch bei Grafikprogrammen zeigt sich der »elite« SP 120VC von seiner besten Seite, mit 480 Punkte/Zeile. Voll Software-kompatibel zum Commodore MPS 801.

Anschlußfertig an Commodore VC 20, C 64, C 128 (D), C 16, C 116, plus/4. Inkl. Dokumentation.

Klare Sache: Noch heute bestellen oder in die nächste Kaufhof-Filiale gehen!

Bestellschein:

bitte einsenden an: Kaufhof AG, KE 614-Bürotechnik, Postfach 10 10 08, 5000 Köln 1

Ich bestelle:

.... Stück »elite« SP-120VC
Matrixdrucker für Commodore-Heimcomputer
Preis je DM 399,—

.... Stück Traktor für »elite« SP-120VC
ermöglicht problemlosen Transport von Computerendlospapier und Etiketten
Preis je DM 39,95

Lieferung per Nachnahme

Name/Vorname: ____________________

Straße/Hausnummer: ____________________

PLZ/Ort: ____________________

Postkarte mit 60 Dpf., Brief mit 80 Dpf. frankieren! 64S24

Ein Beispielprogramm für eine Sinuskurve:

```
10 HGR 0    : REM Grafik ein (gesplittet)
20 GCLR     : REM Grafik löschen
30 SET 3,0  : REM Farbe setzen
40 HOME 1   : REM Textbildschirm löschen
50 PRINT "          Sinuskurve "
60 FOR X = 0 TO 319
70 PLOT X,SIN(X/5)*20+100
80 NEXT
90 GET A$ : IF A$ = "" THEN 90
100 HOME 0 : REM Bildschirm löschen
110 TEXT   : REM Auf Textbildschirm umschalten
120 END
```

LINE X1,Y1,X2,Y2

LINE zeichnet eine Linie im Grafikbildschirm. Die Koordinaten X1 und Y1 geben den Anfangspunkt und die Koordinaten X2 und Y2 geben den Endpunkt der Linie an. Wie beim PLOT Befehl gelten auch hier die Einschränkungen der Koordinaten:

$0 \leqq X \leqq 319$

$0 \leqq Y \leqq 199$

HLINE X, Y, Länge

HLINE gehört mit zu den Befehlen, die noch nie in einer Grafikerweiterung vorhanden waren. Dieser Befehl zieht eine horizontale Linie in einer unvorstellbaren Geschwindigkeit. Wenn man sich überlegt, daß viele Linien gerade verlaufen, kann man es sich zunutze machen, daß die Bitmap so angelegt ist, daß ein gesetztes Byte einer horizontalen Linie von 8 Punkten entspricht. Es wird der Anfangspunkt berechnet und die Länge der Linie durch acht geteilt. Jetzt wird zur Anfangsadresse acht addiert und dieses Byte ganz gesetzt. Dieser Vorgang wird so oft wiederholt, bis die Anzahl der Schleifendurchläufe genau INT(Länge/8) beträgt. Die restlichen Punkte, das vordere und hintere Stück, werden mit der normalen LINE-Routine gesetzt. Dieser Algorithmus hat den Vorteil, daß nicht jeder Punkt berechnet und gesetzt werden muß, wie es bei den üblichen Erweiterungen der Fall ist. Die Parameter X und Y geben den Anfangspunkt an; die Länge ist natürlich die Länge der Linien in Punkten gezählt. Man sollte aber auf jeden Fall folgende Bedingung einhalten:

$X + \text{Länge} \leqq 319$

VLINE X, Y, Länge

VLINE ist auch ein ganz neuer Befehl, der HLINE sehr ähnlich ist. VLINE zieht aber im Gegensatz zu HLINE vertikale Linien, und das mit derselben Geschwindigkeit wie HLINE. Es wird auch hier der Anfangspunkt aus X und Y berechnet. Danach wird die Position in Byte errechnet und der Punkt gesetzt. Addiert man nun den Wert 1, so erhält man den nächsten Punkt, den man setzen muß. Wenn ein 8-Byte-Block zu Ende ist, addiert man 312, um zum nächsten 8-Byte-Block zu gelangen. Für diesen Befehl muß die Bedingung $Y + \text{Länge} \leqq 199$ eingehalten werden.

BLOCK X, Y, Länge, Breite

Es wird ein ausgefülltes Rechteck gezeichnet. X und Y geben die linke obere Ecke an, gefolgt von der Länge und Breite, die die Größe des Blocks bestimmen. Da dieser Befehl auch die Routine HLINE benutzt, sollte man die Parameter mit Sorgfalt behandeln. Dies ist übrigens einer der schnellsten Block-Befehle für den Commodore 64, denn er setzt zirka 128000 (!) Punkte pro Sekunde; 64000 Punkte (ein Bildschirm voll) setzt die Routine in $^{27}/_{60}$ (= $^{9}/_{20}$) Sekunden. Sie können es ja einmal ausprobieren mit BLOCK 0,0,319,199.

REC X, Y, Länge, Breite

Der REC-Befehl ist mit BLOCK gleichzusetzen; er hat die gleiche Syntax, zeichnet aber nur Verbindungslinien des Rechtecks.

CIRCLE X, Y, X-Ausdehnung, Y-Ausdehnung

Es werden beliebige Ellipsen gezeichnet. X und Y sind die Koordinaten des Mittelpunkts; die Y-Ausdehnung ist

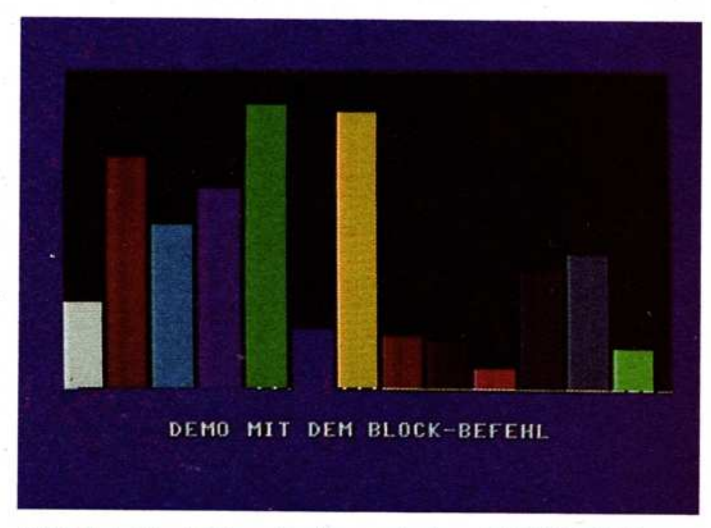


Bild 6. Farbige Balken-Grafiken mit dem BLOCK-Befehl

der Radius in der Y-Achse. Genauso verhält es sich bei der X-Ausdehnung; sie entspricht dem Radius in der X-Richtung.

INK alte Farbe, neue Farbe

Mit dem INK-Befehl kann man aus dem sonst einfarbigen Grafikbildschirm einen farbigen gestalten. Die Auflösung

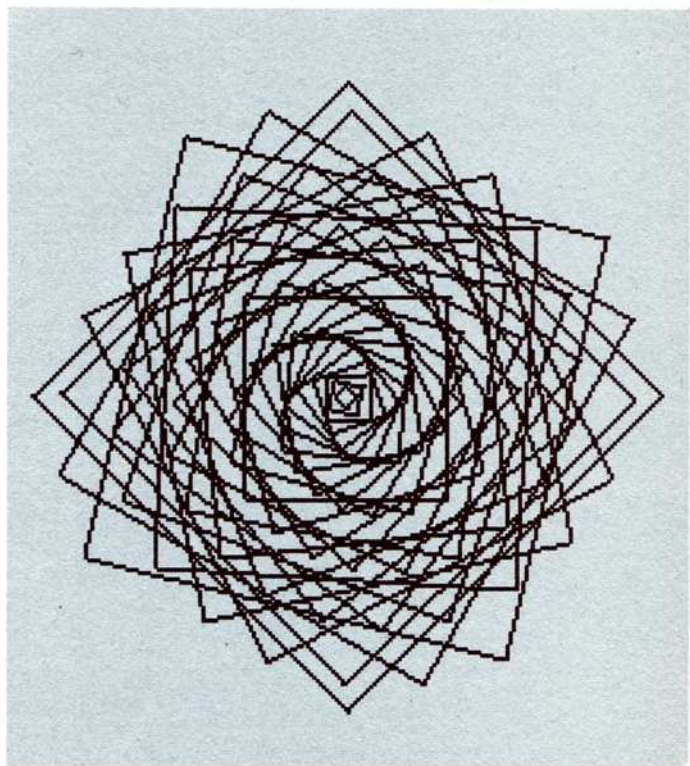


Bild 7. Eine Kreation nur mit dem REC-Befehl

von 320 x 200 Punkten bleibt dennoch erhalten. Es wird in jedem 8-Byte-Block geprüft, ob ein Punkt gesetzt ist. Wenn nein, dann wird der Wert, der im Video-RAM steht, geändert. Ist jedoch ein Punkt gesetzt, bleibt der alte Farbwert erhalten.

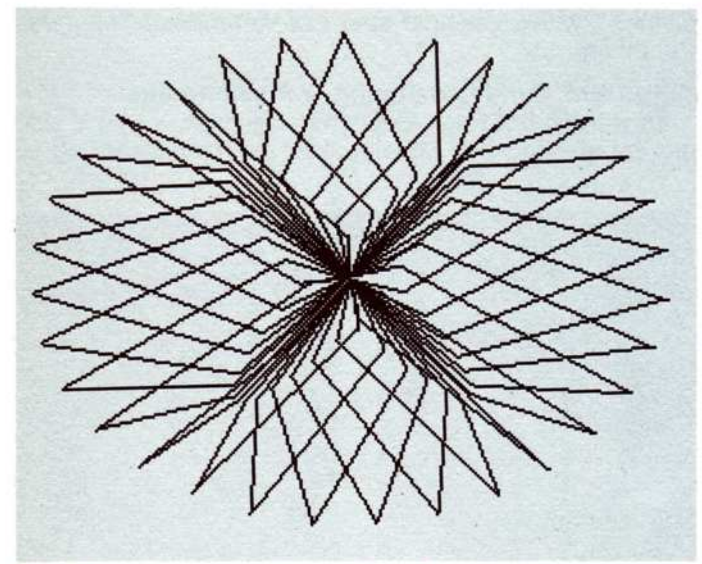


Bild 8. Auch Hardcopies lassen sich erzeugen

COPY

Es wird eine Hardcopy des Grafikbildschirms ($6000-$7FFF) auf den Drucker ausgegeben. Für die Besitzer eines Userport-Centronic-Interface ist ein Treiber vorhanden, der mit »CEN« aktiviert wird. Bei der Verwendung des COPY-Befehls ist darauf zu achten, daß der Grafikbildschirm ausgeschaltet ist, da sonst eine Fehlfunktion auftritt.

CHANGE A

Wenn A den Wert 0 enthält, dann wird der Grafikbereich von $6000 bis $7FFF mit dem Bereich von $A000 bis $BFFF vertauscht. Bei A = 1 wird der Bereich ab $6000 mit dem ab $A000 ODER verknüpft und danach bei $6000 abgelegt.

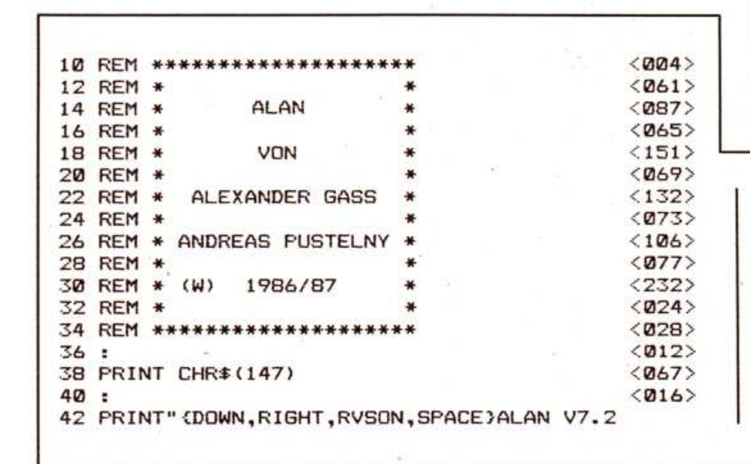
GSAVE "Filename"

GSAVE speichert den Bereich von $5C00 bis $8000 auf Diskette unter dem Namen Filename. Es wird so das Videoram ($5C00 bis $5FFF) und die Bitmap ($6000 bis $7FFF) gespeichert.

Dies waren alle neuen Befehle, die nicht nur jenem Basic-Programmierer zugute kommen, der sich mit der zeitkritischen hochauflösenden Grafik auseinandersetzt.

Sie können auch Adventures, Kalkulationsprogramme und vieles mehr mit »Alan V7.2« programmieren.

(A. Pustelny/A. Gaß/Th. Lipp)



```
10 REM ********************           <004>
12 REM *                  *           <061>
14 REM *       ALAN       *           <087>
16 REM *                  *           <065>
18 REM *       VON        *           <151>
20 REM *                  *           <069>
22 REM *  ALEXANDER GASS  *           <132>
24 REM *                  *           <073>
26 REM * ANDREAS PUSTELNY *           <106>
28 REM *                  *           <077>
30 REM * (W)  1986/87     *           <232>
32 REM *                  *           <024>
34 REM ********************           <028>
36 :                                  <012>
38 PRINT CHR$(147)                    <067>
40 :                                  <016>
42 PRINT"{DOWN,RIGHT,RVSON,SPACE}ALAN V7.2
   WIRD GELADEN ! "                   <158>
44 :                                  <020>
46 IF A=0 THEN A=1:LOAD "ALAN V7.2",8,1   <024>
48 IF A=1 THEN A=2:LOAD "ALAN II",8,1     <091>
50 :                                  <026>
52 SYS 50672 : REM * ALAN AKTIVIEREN *    <014>
54 :                                  <030>
```

Listing 1. »Lader Alan V7.2«, bitte mit dem Checksummer (Seite 159) eingeben.

```
Name : alan v7.2            c000 cf90
----------------------------------------
c000 : a2 45 a0 c1 8e 04 03 8c   d3
c008 : 05 03 a2 08 a0 c2 8e 06   9f
c010 : 03 8c 07 03 a2 3d a0 c2   98
c018 : 8e 08 03 8c 09 03 a9 00   4c
c020 : 85 37 85 33 a9 90 85 38   ae
c028 : 85 34 60 43 41 d4 42 4c   a4
c030 : 4f 41 c4 43 43 4f 50 d9   5d
c038 : 43 4e 45 d5 43 41 4c d4   c7
c040 : 48 45 4c d0 48 47 d2 49   f5
c048 : 4e d6 47 43 4c d2 54 45   73
c050 : 58 d4 53 54 41 54 55 d3   25
c058 : 44 49 53 cb 54 52 41 4e   09
c060 : 53 46 45 d2 57 47 45 d4   f0
c068 : 43 4f cc 56 54 41 c2 53   52
c070 : 45 d4 4c 4c 49 53 d4 46   cb
c078 : 49 4e c4 4f 4c c4 47 53   b2
c080 : 41 56 c5 43 48 41 4e 47   1c.
c088 : c5 44 45 cc 44 4d 4f 44   cf
c090 : c5 44 4f 46 c6 53 43 52   cd
c098 : 4f 4c cc 43 45 ce 41 55   23
c0a0 : 54 cf 49 4e cb 57 49 4e   31
c0a8 : 4f ce 57 49 4e 4f 46 c6   63
c0b0 : 50 4c 4f d4 48 4c 49 4e   3d
c0b8 : c5 56 4c 49 4e c5 42 4c   99
c0c0 : 4f 43 cb 52 45 c3 43 49   00
c0c8 : 52 43 4c c5 48 4f 4d c5   47
c0d0 : 4c 53 43 52 4f 4c cc 52   10
c0d8 : 45 ce 4d 4f 44 c5 4c 49   f8
c0e0 : 4e c5 43 4f 50 d9 00 63   66
c0e8 : c2 ec c2 07 c3 32 c3 38   ff
c0f0 : c3 3e c3 8e c3 0d c4 29   9f
c0f8 : c4 40 c4 72 c4 9b c4 d5   44
c100 : c4 0f c5 9f c5 b8 c5 c8   7c
c108 : c5 b8 c6 f3 c6 d8 c7 14   d4
c110 : c8 29 c8 83 c8 4b c9 82   22
c118 : c9 8d c9 34 ca 21 ce c6   1f
c120 : ce 86 cf df 99 b3 ca 19   b6
c128 : cb bb cb 16 cc 52 cd 2f   7b
c130 : 92 b4 ce ff 96 e4 cd 1f   d6
c138 : cf ff 8f 53 91 00 60 1d   2a
c140 : 91 9d 93 11 13 a6 7a a0   38
c148 : 04 84 0f bd 00 02 10 07   68
c150 : c9 ff f0 3e e8 d0 f4 c9   99
c158 : 20 f0 37 85 08 c9 22 f0   a8
c160 : 55 24 0f 70 2d c9 3f d0   59
c168 : 04 a9 99 d0 25 c9 30 90   44
c170 : 04 c9 3c 90 1d 84 71 a0   77
c178 : 00 84 0b 88 86 7a ca c8   87
c180 : e8 bd 00 02 38 f9 9e a0   96
c188 : f0 f5 c9 80 d0 2f 05 0b   a6
c190 : a4 71 e8 c8 99 fb 01 c9   51
c198 : 00 f0 38 38 e9 3a f0 04   62
c1a0 : c9 49 d0 02 85 0f 38 e9   08
c1a8 : 55 d0 a0 85 08 bd 00 02   b1
c1b0 : f0 e0 c5 08 f0 dc c8 99   cf
c1b8 : fb 01 e8 d0 f0 a6 7a e6   84
c1c0 : 0b c8 b9 9d a0 10 fa b9   3b
c1c8 : 9e a0 d0 b5 f0 0f bd 00   20
c1d0 : 02 10 bd 99 fd 01 c6 7b   77
c1d8 : a9 ff 85 7a 60 a0 00 b9   b0
c1e0 : 2b c0 d0 02 c8 e8 bd 00   aa
c1e8 : 02 38 f9 2b c0 f0 f5 c9   e9
c1f0 : 80 d0 04 05 0b d0 99 a6   65
c1f8 : 7a e6 0b c8 b9 2a c0 10   d1
c200 : fa b9 2b c0 d0 e0 f0 c6   1f
c208 : 10 0f 24 0f 30 0b c9 ff   0d
c210 : f0 07 c9 cc b0 06 4c 24   44
c218 : a7 4c f3 a6 38 e9 cb aa   0e
c220 : 84 49 a0 ff ca f0 08 c8   57
c228 : b9 2b c0 10 fa 30 f5 c8   44
c230 : b9 2b c0 30 05 20 47 ab   7b
c238 : d0 f5 4c ef a6 20 73 00   4d
c240 : 20 46 c2 4c ae a7 c9 cc   a6
c248 : 90 04 c9 fe 90 06 20 79   d9
c250 : 00 4c ed a7 38 e9 cc 0a   01
c258 : aa bd e8 c0 48 bd e7 c0   c7
c260 : 48 4c 73 00 a9 24 8d ec   77
c268 : c2 a9 01 a2 ec a0 c2 20   b3
c270 : bd ff a9 08 aa a0 60 20   0a
c278 : ba ff 20 c0 ff a5 ba 20   aa
c280 : b4 ff a5 b9 20 96 ff a9   de
c288 : 00 85 90 a0 03 8c ec c2   51
c290 : 20 a5 ff 85 fc a4 90 d0   0c
c298 : 42 20 a5 ff a4 90 d0 3b   dc
c2a0 : ac ec c2 88 d0 e7 a6 fc   65
c2a8 : 20 cd bd a9 20 20 d2 ff   a2
c2b0 : a0 00 20 a5 ff c9 22 f0   c6
c2b8 : 27 a6 90 d0 1e aa f0 12   90
c2c0 : 20 d2 ff c8 20 ed f6 f0   91
c2c8 : 12 a5 c5 c9 3c f0 f5 4c   13
c2d0 : b2 c2 a9 0d 20 d2 ff a0   c9
c2d8 : 02 d0 b2 a9 08 4c c3 ff   16
c2e0 : c0 05 90 d5 a9 22 20 d2   d3
c2e8 : ff 4c c4 c2 00 20 9e ad   6e
c2f0 : 20 a3 b6 20 bd ff a9 08   26
c2f8 : aa a8 20 ba ff a5 00 20   c3
c300 : 90 ff a9 00 20 d5 ff 60   6c
c308 : 78 a5 01 48 29 fb 85 01   27
c310 : a9 d0 85 03 a9 30 85 05   1f
c318 : a0 00 84 02 84 04 a2 20   4d
c320 : b1 02 91 04 c8 d0 f9 e6   80
c328 : 03 e6 05 ca d0 f2 68 85   8a
```

Listing 2. »Alan V7.2«, eine Basic-Erweiterung mit vielen neuen Befehlen. Bitte mit dem MSE (Seite 159) eingeben.

Faszinierende Spielewelt

Laß' Dich in eine abenteuerliche Spielewelt entführen:

**alles, was Du dazu brauchst, ist ein C64 oder ein C128, die Spieledisketten – und schon kann die Reise losgehen!
Nutze Deine Joystick-Künste, indem Du sicher einen Weg aus dem Labyrinth findest. Bewahre Deinen kühlen Kopf in aufregenden Aktionszenen und Kampfduellen! Zeige Deine Fähigkeiten als Börsenmakler in lebensnahen Wirtschaftssimulationen! Beweise Deine Fähigkeiten als Präsident, Manager und Trainer eines Bundesligaclubs und und und…**

Die 64'er-Spielesammlung, Band 1, 1987, 115 Seiten, inklusive Diskette



Mit den 15 spannenden Spielen, der ausführlichen Anleitung sowie den farbigen Bildschirmfotos ist Dir ein fantastisches Spielvergnügen gewiß:
Balliard: Einfallswinkel = Ausfallswinkel. Wer das nicht befolgt, hat es schwer bei dieser Mischung aus Tennis und Billard. **The Way:** Zu verschlungenen Pfaden gesellen sich Geldsäcke und böse Geister, die es zu bekämpfen gilt. **Vager 3**: Joystickprofis mit ungetrübtem Visierblick und Trefferinstinkt können ihr Punktekonto schwer mit Abschußprämien beladen. **Firebug**: Hoffentlich fängt Dein Joystick nicht ebenfalls Feuer, wenn es heißt, die wertvollen Koffer aus dem brennenden Haus des Professors zu erwischen. **Pirat**: Taktik, Timing und gute Navigationskenntnisse sind Voraussetzung für ein bis zu 25 Jahre langes Piratenleben. **Wirtschaftsmanager**: Simulation aus den höchsten Etagen der Wirtschaft, nicht 1000 Stück, sondern ganze Firmen gehen über den »Ladentisch«. **Vier gewinnt**: Einfach, aber gerade deshalb ein Spiel, das schnell zu Erfolgserlebnissen führt. **Brainstorm**: Mastermind stand Pate für dieses vielseitige Denkspiel. **Hypra-Chess**: Spiel' Schach gegen einen C64. **Maze:** Wer die Übersicht behält und nicht kopflos herumspringt, wird das Ziel erreichen. **Schiffe versenken:** Endlich eine faire Version dieses weitverbreiteten Spieles, mit zusätzlichen Spielvarianten. **Handel:** Hier kannst Du deinen Geschäftssinn und Deine Risikobereitschaft unter Beweis stellen, ohne wirklich später am Hungertuch nagen zu müssen. **Börse:** Diese Spiel bietet wirklich einen hervorragenden Einstieg ins Börsenkarussell. Außerdem sind noch die Spiele **Vier in vier** und **Magic-Cubs** enthalten.

Hardware-Anforderungen:
C64 oder C128 bzw. C128D (64er-Modus), Floppy 1541, 1570 oder 1571 und Joystick.

Bestell-Nr. 90429, ISBN 3-89090-429-7

DM 39,–* (sFr 35,90*/öS 304,20*)

64'er Extra Nr. 4, Abenteuerspiele, 1987, 17 Seiten, drei Disketten

Robox: Fesselndes Grafik-Science-Fiction-Adventure. Der Herrscher eines fremden Planeten ließ sein Gehirn nach seinem Tod künstlich weiterleben – in einem Körper ohne Seele. Aus dieser Kombination, halb Roboter und halb Mensch, halb lebend und halb tot, die Robox genannt wurde, enstand der Haß auf alles Lebende. Er befahl daher seiner Roboter-Armee, sämtliches Leben des Universums zu vernichten. Deine Aufgabe ist es, zu Robox zu gelangen und ihn unschädlich zu machen, um die Erde vor ihrem sonst sicheren Tod zu bewahren. Wie Du das tust, bleibt Dir überlassen. **Scotland Yard:** Spannendes Kriminal-Adventure. Begib Dich auf spannende Verbrecherjagd in das London des 19. Jahrhunderts, und lasse Dich engagieren bei Scotland Yard. Verhöre Tatverdächtige, prüfe deren Alibis und verfolge die Spuren zurück zum Täter. Als Belohnung für die Lösung der zehn ungeklärten Fälle wartet Deine Beförderung zum Oberinspektor. Mit dem mitgelieferten Fall-Editor konstruierst Du weitere Verbrechen und gibst damit Deinen Freunden harte Nüsse zu knacken.

3 Disketten (beidseitig bespielt) für den C64/C128

Bestell-Nr. 38704

DM 29,90* (sFr 24,90*/öS 299,–*)

Außerdem gibt es in der Software-Reihe »64'er Extra« noch folgende Programme: Best of Grafik Vol. 1, 3D-Grafik für C64, Grafik-Design, Tips & Tricks, Bestell-Nr. 38701, DM 49,90* (sFr 44,90*/öS 499,–*) • Best of Grafik Vol. 2, Scrolling für Spiele, Fractal-Landschaften, Business-Grafik, Grafik-Erweiterungen, Super-Drucker-Software, Bestell-Nr. 38702, DM 39,90* (sFr 34,90*/öS 399,–*) • Best of Grafik Vol. 3, Erweiterungen für Grafik und Spiele, 3D-Trickfilm, Apfelmännchen, Super-Hardcopies, Bestell-Nr. 38703, DM 39,90* (sFr 34,90*/öS 399,–*)

Die 64'er-Spielesammlung, Band 2, 1987, 98 Seiten, inklusive Diskette

Auch der zweite Band der Spielesammlung mit 14 spannenden Spielen entführt Dich in eine fantastische Action-Welt:
Billard: Banden- oder Lochbillard, allein oder zu zweit, das ist hier die eine Frage. Wie Du die Kugeln dann einlochst, eine ganz andere. **Tonti:** Ein schnelles Auge und geschickte Hände am Joystick werden verlangt, um auch einmal eine Tontaube zu treffen. **Freiheit:** Du bist auf einer Insel, gefangen im untersten Verlies. Wie flüchtest Du zur Nachbarinsel? Der Weg wird schwierig, denn bis zu hundert Räume sind zu meistern. Ein Textadventure besonderer Klasse. **Apokalypse Now:** Ein Hubschrauber-Klassiker, ein Spiel mit tödlichem Ausgang – nicht unbedingt für Dich – aber nur, wenn Du am Joystick Profiqualitäten vorweisen kannst **Black out:** Ein Spielhallenhit für den C64. Reaktionsvermögen, eine ruhige Hand und ein bißchen Glück gehören schon dazu, um alle Blöcke abzuschießen. **Aquantor:** Wasser, Wasser, Wasser, so weit der Bildschirm reicht. Nur schnell einen Damm errichten und fleißig punkten, wenn man im Trockenen sitzt. **Tödliches Dioxin**: Du hast vier hochmoderne Hubschrauber und bist selbst ein wagemutiger, unerschrockener Pilot. Der Haken an der Sache ist die Fracht – tödliches Dioxin. **Libra:** Du fliegst für die intergalaktische Föderation der Raumritter und rettest ein unabhängiges Sonnensystem. **Dasher:** Pac-Man stand Pate für dieses Spiel. Zu den bekannten Funktionen, wie Punkte fressen und Gegener vernichten, erhältst Du die Möglichkeit, eigene Spielfelder zu entwerfen und aufzubauen. **Bundesliga Manager:** Einer der 18 Bundesligaclubs wartet auf Dein Multitalent und Du hast wirklich mächtige Mittel zur Hand, um erfolgreich zu agieren. Außerdem sind noch die Spiele **Golf, Zauberschloß, Steel Slab** und **Space Invader** enthalten.

Hardware-Anforderungen für Band 1 und Band 2:
C64 oder C 128 bzw. C128D (64'er Modus), Floppy 1541, 1570 oder 1571 und Joystick.

Bestell-Nr. 90428, ISBN 3-89090-428-9

DM 39,–* (sFr 35,90*/öS 304,20*)

* Unverbindliche Preisempfehlung



Markt & Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften oder in den Fachabteilungen der Warenhäuser.

Markt & Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.

```
c330 : 01 58 60 a9 1d 8d 18 d0   eb
c338 : 60 a9 15 8d 18 d0 60 a9   41
c340 : 0d 20 d2 ff a2 00 86 9e   93
c348 : 86 9b a6 d6 86 9f a4 9e   55
c350 : 84 d3 20 10 e5 a6 9b bd   45
c358 : 2b c0 c9 00 d0 05 a9 0d   4c
c360 : 4c d2 ff a8 29 80 c9 80   e9
c368 : f0 09 98 20 d2 ff e6 9b   07
c370 : 4c 52 c3 98 29 7f 20 d2   9e
c378 : ff e6 9b a5 9e 69 0a c9   77
c380 : 27 90 07 a9 0d 20 d2 ff   03
c388 : a9 00 85 9e 4c 4e c3 20   ed
c390 : 9e b7 e0 01 f0 63 78 a9   c1
c398 : b5 a0 c3 8d 14 03 8c 15   f6
c3a0 : 03 a9 14 8d 12 d0 ad 11   af
c3a8 : d0 29 7f 8d 11 d0 a9 81   e0
c3b0 : 8d 1a d0 58 60 ad 19 d0   03
c3b8 : 8d 19 d0 30 07 ad 0d dc   d8
c3c0 : 58 4c 31 ea ad 12 d0 c9   2a
c3c8 : d1 b0 17 a9 3b 8d 11 d0   f2
c3d0 : a9 78 8d 18 d0 a9 d1 8d   d8
c3d8 : 12 d0 a9 02 8d 00 dd 4c   e6
c3e0 : bc fe a9 1b 8d 11 d0 a9   e1
c3e8 : 15 8d 18 d0 a9 03 8d 00   cd
c3f0 : dd a9 14 8d 12 d0 4c bc   ab
c3f8 : fe a9 00 8d 1a d0 a9 02   4f
c400 : 8d 00 dd a9 3b 8d 11 d0   40
c408 : a9 78 8d 18 d0 60 a9 60   cb
c410 : 85 9c a9 00 85 9b a0 00   05
c418 : b1 9b 49 ff 91 9b c8 d0   a4
c420 : f7 e6 9c a4 9c c0 80 d0   b9
c428 : ed 60 a9 60 85 9c a9 00   a0
c430 : 85 9b a8 91 9b c8 d0 fb   1a
c438 : e6 9c a4 9c c0 80 d0 f2   62
c440 : 60 a9 00 8d 1a d0 78 a9   84
c448 : 31 a0 ea 8d 14 03 8c 15   eb
c450 : 03 58 a9 15 8d 18 d0 a9   bd
c458 : 1b 8d 11 d0 a9 03 8d 00   81
c460 : dd a9 c8 8d 16 d0 a0 00   60
c468 : a9 20 99 c0 07 c8 c0 28   aa
c470 : d0 f8 60 a9 01 a2 08 a0   90
c478 : 0f 20 ba ff a9 00 20 bd   dc
c480 : ff 20 c0 ff a2 01 20 c6   00
c488 : ff 20 cf ff 20 d2 ff 24   6c
c490 : 90 50 f6 20 cc ff a9 01   7f
c498 : 20 c3 ff 60 20 9e ad 20   94
c4a0 : a3 b6 85 03 86 04 84 05   05
c4a8 : a9 02 a2 08 a0 0f 20 ba   74
c4b0 : ff a9 00 20 bd ff 20 c0   66
c4b8 : ff a2 02 20 c9 ff a0 00   ac
c4c0 : b1 04 20 d2 ff c8 c4 03   35
c4c8 : d0 f6 20 cc ff 20 73 c4   0d
c4d0 : a9 02 20 c3 ff 60 20 9a   b4
c4d8 : ad 20 a3 b6 86 9b 84 9c   e6
c4e0 : aa a0 00 ca b1 9b 99 3e   0f
c4e8 : 03 c8 e0 00 d0 f5 98 18   d7
c4f0 : 69 3e 8d f8 c4 a9 20 8d   30
c4f8 : 3e 03 ee f8 c4 ac f8 c4   b2
c500 : c0 ec d0 f3 8d 3c 03 8d   cb
c508 : 3d 03 a9 0e 20 d2 ff 60   4c
c510 : a9 00 85 b4 a4 d3 b1 d1   04
c518 : 85 9b b1 f3 85 9c a9 a4   83
c520 : 20 d2 ff a9 9d 20 d2 ff   04
c528 : 20 e4 ff f0 fb a2 06 dd   81
c530 : 3e c1 f0 f4 ca d0 f8 c9   d4
c538 : 0d f0 03 4c 50 c5 a4 d3   75
c540 : a5 9b 91 d1 a5 9c 91 f3   bf
c548 : a9 0d 20 d2 ff 4c 6a c5   72
c550 : 20 d2 ff c9 14 f0 03 4c   80
c558 : 5f c5 c6 b4 4c 14 c5 a6   ac
c560 : b4 9d 3c 03 e8 86 b4 4c   80
c568 : 14 c5 a6 b4 8a 20 7d b4   a8
c570 : a0 00 b9 3c 03 91 62 c4   d6
c578 : 61 f0 03 c8 d0 f4 20 8b   77
c580 : b0 a6 0d f0 18 85 b5 84   72
c588 : b6 a2 02 a0 02 b5 61 91   9a
c590 : b5 ca 88 10 f8 a2 00 86   80
c598 : cf e8 86 cc 60 4c 99 ad   41
c5a0 : 20 9e b7 8e 20 d0 20 fd   d4
c5a8 : ae 20 9e b7 8e 21 d0 20   7a
c5b0 : fd ae 20 9e b7 8e 86 02   ee
c5b8 : 60 20 9e b7 e0 19 b0 06   6c
c5c0 : 86 d6 20 6c e5 60 4c 08   ea
c5c8 : af 20 9e b7 8a 0a 0a 0a   5b
c5d0 : 0a 85 9b 20 fd ae 20 9e   9b
c5d8 : b7 8a 05 9b a0 00 99 00   fa
c5e0 : 5c 99 fa 5c 99 f4 5d 99   3d
c5e8 : ee 5e c8 c0 fa d0 ef 60   06
c5f0 : a9 25 a2 c6 8d 00 80 8e   a5
c5f8 : 01 80 a0 fe a9 5e 8d 02   09
c600 : 80 8c 03 80 a9 c3 a2 c2   60
c608 : a0 cd 8d 04 80 8e 05 80   04
c610 : 8c 06 80 a9 38 a2 30 8d   69
c618 : 07 80 8e 08 80 a9 ed 8d   2c
c620 : 28 03 4c e2 fc 20 a3 fd   95
c628 : 20 8e a6 20 44 a6 20 53   de
c630 : e4 20 bf e3 20 a3 fd 20   e8
c638 : 5e ff 20 00 c0 20 46 c6   52
c640 : 20 41 c4 4c 74 a4 a9 62   93
c648 : a0 c6 20 1e ab a5 37 38   4c
c650 : e5 2b aa a5 38 e5 2c 20   ce
c658 : cd bd a9 8c a0 c6 20 1e   fd
c660 : ab 60 93 11 1d 1d 1d 1d   ac
c668 : 41 4c 41 4e 20 56 37 2e   d7
c670 : 32 20 42 59 20 41 2e 47   c1
c678 : 41 53 53 20 26 20 41 2e   01
c680 : 50 55 53 54 45 4c 4e 59   7d
c688 : 0d 11 1d 00 13 11 11 11   85
c690 : 1d 1d 1d 1d 1d 1d 1d 42   db
c698 : 41 53 49 43 20 42 59 54   60
c6a0 : 45 53 20 46 52 45 45 20   04
c6a8 : 20 36 34 4b 20 52 41 4d   8e
c6b0 : 20 53 59 53 54 45 4d 0d   f9
c6b8 : 00 a9 04 aa 20 ba ff 20   fb
c6c0 : 4a f3 a2 04 20 c9 ff a9   d1
c6c8 : d7 a2 c6 8d 00 03 8e 01   a8
c6d0 : 03 20 79 00 4c 9c a6 a9   d9
c6d8 : 8b a2 e3 8d 00 03 8e 01   b3
c6e0 : 03 a9 0a 20 d2 ff a9 04   1a
c6e8 : 20 c3 ff 20 cc ff 4c 74   d5
c6f0 : a4 4c 08 af a9 af a2 c7   e4
c6f8 : 8d 00 03 8e 01 03 a5 2b   2d
c700 : a6 2c 85 a9 86 aa a2 00   9b
c708 : bd 00 02 e8 c9 de d0 f8   2c
c710 : a0 ff c8 bd 00 02 e8 99   81
c718 : 3c 03 c9 00 d0 f4 88 84   28
c720 : 05 20 bf c7 a2 00 a0 00   cb
c728 : a1 a9 48 d0 09 20 8a c7   15
c730 : 20 bf c7 4c 39 c7 20 8a   13
c738 : c7 68 d9 3c 03 f0 22 a5   bd
c740 : aa c5 2e 90 df f0 02 b0   59
c748 : 06 a5 a9 c5 2d 90 d5 a9   46
c750 : 8b a2 e3 8d 00 03 8e 01   2b
c758 : 03 a9 0d 20 d2 ff 4c 74   be
c760 : a4 c4 05 d0 17 c8 84 02   90
c768 : a5 a9 85 03 38 e5 02 85   69
c770 : a9 a5 aa 85 04 e9 00 85   e2
c778 : aa 4c 91 c7 a1 a9 c8 20   70
c780 : 8a c7 d9 3c 03 d0 b8 4c   1e
c788 : 61 c7 e6 a9 d0 02 e6 aa   ca
c790 : 60 a5 b4 18 69 02 85 a9   03
c798 : a5 b5 69 00 85 aa a0 00   a2
c7a0 : b1 a9 85 14 c8 b1 a9 85   d5
c7a8 : 15 20 13 a6 4c bd a6 a5   00
c7b0 : 03 85 a9 a5 04 85 aa c6   39
c7b8 : d6 20 10 e5 4c 3f c7 a5   88
c7c0 : a9 85 b4 18 69 04 85 a9   7c
c7c8 : a5 aa 85 b5 69 00 85 aa   dc
c7d0 : 20 ed f6 f0 01 60 4c 4f   a5
c7d8 : c7 a5 2c 85 03 a5 2b 85   43
c7e0 : 02 a0 03 c8 b1 2b d0 fb   bc
c7e8 : c8 18 98 65 2b a0 00 91   6a
c7f0 : 2b c8 a5 2c 69 00 91 2b   a1
c7f8 : a0 00 b1 02 c8 aa b1 02   f2
c800 : f0 06 85 03 86 02 d0 f0   53
c808 : a5 02 69 02 85 2d a5 03   a7
c810 : 69 00 85 2e 60 20 d4 e1   bf
c818 : a9 00 aa 85 fb a9 5c 85   a6
c820 : fc a0 80 a9 fb 85 b9 4c   2d
c828 : d8 ff 20 9e b7 86 02 a9   e7
c830 : 36 85 01 a9 5c 85 fc a9   d8
c838 : 9c 85 fe a9 00 85 fb 85   b3
c840 : fd a0 00 a5 02 f0 06 20   42
c848 : 78 c8 4c 50 c8 20 6d c8   16
c850 : e6 fb e6 fd d0 04 e6 fc   70
c858 : e6 fe a5 fc c9 80 d0 e3   72
c860 : a9 37 85 01 a9 00 85 33   3d
c868 : a9 a0 85 34 60 b1 fb aa   22
c870 : b1 fd 91 fb 8a 91 fd 60   f1
c878 : b1 fd 11 fb 91 fb 60 a2   ab
c880 : 11 4c 37 a4 90 06 f0 f7   07
c888 : c9 ab d0 f3 20 6b a9 20   1e
c890 : 13 a6 b0 06 a5 14 05 15   1c
c898 : d0 e5 a5 5f a4 60 85 02   17
c8a0 : 84 25 20 79 00 f0 0c c9   39
c8a8 : ab d0 d4 20 73 00 20 6b   83
c8b0 : a9 d0 cc 68 68 a5 14 05   10
c8b8 : 15 d0 31 a5 2d 38 e9 02   77
c8c0 : 85 fb 85 fd a5 2e e9 00   d7
c8c8 : 85 fc 85 fe a0 00 b1 fd   d9
c8d0 : c9 00 d0 6b c8 b1 fd c5   d8
c8d8 : fb d0 64 c8 b1 fd c5 fc   8a
c8e0 : d0 5d c8 b1 fd 85 14 c8   b5
c8e8 : b1 fd 85 15 20 13 a6 90   f2
c8f0 : 8e a0 00 b1 5f 85 04 c8   c8
c8f8 : b1 5f 85 23 a5 2d 85 22   3d
c900 : a5 02 38 e5 04 18 65 2d   62
c908 : 85 2d 85 24 a5 2e 69 ff   7b
c910 : 85 2e e5 25 aa 38 a5 02   d1
c918 : e5 2d a8 b0 03 e8 c6 25   b1
c920 : 18 65 22 90 03 c6 23 18   a9
c928 : b1 22 91 24 c8 d0 f9 e6   9c
c930 : 23 e6 25 ca d0 f2 20 59   41
c938 : a6 20 33 a5 4c 74 a4 a9   be
c940 : ff c6 fd c5 fd d0 02 c6   d6
c948 : fe 4c cc c8 a9 57 a0 c9   24
c950 : 8d 02 03 8c 03 03 60 20   3b
c958 : 60 a5 86 7a 84 7b 20 73   07
c960 : 00 aa f0 f3 a2 ff 86 3a   29
c968 : 90 03 4c 96 a4 a9 02 a0   41
c970 : 00 85 7b 84 7a 20 79 00   31
c978 : 20 6b a9 a9 00 8d 00 02   5e
c980 : 4c a2 a4 a9 83 a0 a4 8d   66
c988 : 02 03 8c 03 03 60 a9 01   6b
c990 : 85 9c 85 9b 20 9e b7 8e   2b
c998 : 15 ca 78 a9 b9 8d 14 03   c4
c9a0 : a9 c9 8d 15 03 ad 11 d0   b7
c9a8 : 29 7f 8d 11 d0 a9 c2 8d   97
c9b0 : 12 d0 a9 81 8d 1a d0 58   62
c9b8 : 60 ad 19 d0 8d 19 d0 30   94
c9c0 : 07 ad 0d dc 58 4c 31 ea   ff
c9c8 : a5 9d c9 06 f0 0f a9 06   a9
c9d0 : 85 9d a9 00 8d 12 d0 20   7b
c9d8 : ee c9 4c bc fe a9 00 85   9e
c9e0 : 9d a9 e4 8d 12 d0 a9 c8   1c
c9e8 : 8d 16 d0 4c bc fe 20 27   d0
c9f0 : ca a6 9b ca 86 9b 8e 16   f9
c9f8 : d0 e0 ff f0 01 60 a2 07   02
ca00 : 86 9b 8e 16 d0 a2 00 bd   58
ca08 : c1 07 9d c0 07 e8 e0 27   56
ca10 : d0 f5 a6 9c e0 0a f0 0a   4e
ca18 : e8 86 9c bd 3c 03 8d e7   04
ca20 : 07 60 a9 ff 85 9c 60 a6   ce
ca28 : d6 e0 15 90 04 a2 15 86   7c
ca30 : d6 20 6c e5 60 a2 4a a0   74
ca38 : ca 8e 26 03 8c 27 03 a2   87
ca40 : 9d a0 ca 8e 20 03 8c 21   40
ca48 : 03 60 48 85 9b a5 9a c9   23
ca50 : 04 f0 03 4c cd f1 78 8a   7a
ca58 : 48 98 48 a9 ff 8d 03 dd   68
ca60 : ad 02 dd 09 04 8d 02 dd   17
ca68 : ad 00 dd 09 04 8d 00 dd   16
ca70 : a9 10 8d 0d dd ad 0d dd   62
ca78 : 58 a5 9b 8d 01 dd ad 00   f1
ca80 : dd 29 fb 8d 00 dd 09 04   bd
ca88 : 8d 00 dd ad 0d dd 29 10   c7
ca90 : f0 f9 a9 00 85 90 68 a8   b7
ca98 : 68 aa 68 18 60 20 0f f3   9e
caa0 : f0 03 4c 01 f7 20 1f f3   2a
caa8 : a5 ba c9 04 f0 03 4c 5b   ac
cab0 : f2 4c 75 f2 20 eb b7 20   04
cab8 : f0 ca 8a 4a 4a 4a 85 fd   02
cac0 : 29 03 a8 18 8a 29 07 79   99
cac8 : f4 ca 85 fe a5 14 29 f8   f4
cad0 : 65 fe 85 fe a4 fd b9 f8   08
cad8 : ca 69 60 65 15 85 ff a5   e4
cae0 : 14 29 07 a8 b9 11 cb ea   89
cae8 : ea a0 00 11 fe 91 fe 60   7e
caf0 : 20 68 cc 60 00 40 80 c0   09
caf8 : 00 01 02 03 05 06 07 08   07
cb00 : 0a 0b 0c 0d 0f 10 11 12   0e
cb08 : 14 15 16 17 19 1a 1b 1c   16
cb10 : 1e 80 40 20 10 08 04 02   d8
cb18 : 01 00 a9 00 85 a7 20 eb   72
cb20 : b7 20 68 cc 20 f6 cc a5   d3
cb28 : 14 29 07 49 07 85 ff 20   99
cb30 : fd ae 20 89 cc a5 a9 d0   00
cb38 : 27 a5 ff 38 e5 a8 90 20   5f
cb40 : a5 a8 a8 a5 fd 88 c0 fe   3d
cb48 : f0 0b aa 18 65 a7 85 a7   64
cb50 : 8a 4a 4c 45 cb a2 00 a5   d8
cb58 : a7 ea ea 01 fb 81 fb 60   cc
cb60 : a5 a8 38 e5 ff 85 a8 aa   48
cb68 : a5 a9 e9 00 85 a9 8a 66   f9
cb70 : a9 6a 4a 4a 48 a4 ff 20   14
cb78 : 43 cb a5 a8 29 07 85 aa   55
cb80 : a2 00 68 a8 88 c0 ff f0   c2
cb88 : 14 a9 08 18 65 fb 85 fb   ba
cb90 : a5 fc 69 00 85 fc a9 ff   f5
cb98 : 81 fb 4c 84 cb a5 aa f0   31
cba0 : 1a a8 88 a9 80 85 fd a9   e5
cba8 : 00 85 a7 a9 08 18 65 fb   59
cbb0 : 85 fb a5 fc 69 00 85 fc   e2
cbb8 : 20 43 cb 60 20 a5 cc 20   1b
cbc0 : f6 cc 20 fd ae 20 ae cc   24
cbc8 : a5 fd ea ea 85 fd e8 86   7d
cbd0 : a7 a5 fb 29 07 49 07 a8   96
cbd8 : c0 ff f0 03 4c fb cb a0   49
cbe0 : 07 a9 38 18 65 fb 85 fb   11
```

```
cbe8 : a9 01 65 fc 85 fc a9 01  f4
cbf0 : 18 65 fb 85 fb a5 fc 69  1e
cbf8 : 00 85 fc a2 00 a1 fb ea  21
cc00 : ea 05 fd 81 fb a5 a7 38  18
cc08 : e9 01 85 a7 f0 08 88 c0  bb
cc10 : ff f0 cc 4c ee cb 60 20  53
cc18 : c8 cc a5 14 85 f7 a5 15  0b
cc20 : 85 f8 86 f9 20 fd ae 4c  47
cc28 : cf cc a5 14 85 b0 a5 15  e8
cc30 : 85 05 86 06 a5 f7 85 14  f3
cc38 : a5 f8 85 15 a6 f9 a9 00  3e
cc40 : 85 a7 20 f6 cc a5 14 29  1c
cc48 : 07 49 07 85 ff a5 b0 85  61
cc50 : a8 a5 05 85 a9 20 35 cb  c5
cc58 : a4 06 88 f0 0a 84 06 a6  6a
cc60 : f9 e8 86 f9 4c 3c cc 60  49
cc68 : a5 14 85 b5 a5 15 85 b6  b6
cc70 : e0 c8 90 06 20 41 c4 4c  51
cc78 : 48 b2 a5 b6 c9 01 90 08  50
cc80 : d0 f2 a5 b5 c9 40 b0 ec  25
cc88 : 60 20 8a ad 20 f7 b7 84  fa
cc90 : a8 85 a9 98 18 65 b5 85  07
cc98 : 14 a5 a9 65 b6 85 15 a9  d5
cca0 : 00 aa 4c 68 cc 20 eb b7  02
cca8 : 20 68 cc 86 b7 60 20 9e  3c
ccb0 : b7 8a 18 65 b7 c9 c8 90  6d
ccb8 : 03 4c 74 cc a9 00 69 00  d8
ccc0 : c9 01 90 03 4c 74 cc 60  eb
ccc8 : 20 eb b7 20 68 cc 60 20  7e
ccd0 : eb b7 a5 14 85 b0 a5 15  21
ccd8 : 85 05 e8 86 06 ca a5 14  60
cce0 : 18 65 f7 85 14 a5 15 65  e7
cce8 : f8 85 15 8a 18 65 f9 aa  23
ccf0 : 20 68 cc 4c 34 cc a5 14  69
ccf8 : 29 f8 85 fb a9 00 85 fa  25
cd00 : 85 fe 8a 29 f8 48 0a 26  12
cd08 : fe 0a 26 fe 0a 26 fe 0a  57
cd10 : 26 fe 0a 26 fe 85 ff 68  ea
cd18 : 0a 26 fa 0a 26 fa 0a 26  e4
cd20 : fa 18 65 ff 85 fd a5 fe  5c
cd28 : 65 fa 85 fe 8a 29 07 18  8a
cd30 : 65 fd 65 fb 85 fb a5 15  66
cd38 : 65 fe 69 60 85 fc a5 14  82
cd40 : 29 07 49 07 aa bd 4b cd  81
cd48 : 85 fd 60 01 02 04 08 10  85
cd50 : 20 40 80 20 c8 cc a5 14  66
cd58 : 85 f7 a5 15 85 f8 86 f9  13
cd60 : 20 fd ae 20 eb b7 a5 14  6a
cd68 : 85 b1 a5 15 85 05 86 06  78
cd70 : a5 14 18 65 f7 85 14 a5  19
cd78 : 15 65 f8 85 15 8a 18 65  ff
cd80 : f9 aa 20 68 cc 20 8e cd  87
cd88 : 20 b2 cd 4c c4 cd a5 f7  3f
cd90 : 85 14 a5 f8 85 15 a6 f9  37
cd98 : 20 f6 cc a5 14 29 07 49  54
cda0 : 07 85 ff a9 00 85 a7 a5  b5
cda8 : b1 85 a8 a5 05 85 a9 4c  b6
cdb0 : 35 cb a5 f7 85 14 a5 f8  b5
cdb8 : 85 15 a6 f9 20 f6 cc a6  eb
cdc0 : 06 4c c8 cb a5 f9 85 b0  39
cdc8 : 18 65 06 85 f9 20 8e cd  3c
cdd0 : a5 b0 85 f9 a5 f7 18 65  b3
cdd8 : b1 85 f7 a5 f8 65 05 85  d8
cde0 : f8 20 b2 cd 60 20 79 00  3c
cde8 : 20 eb b7 86 fb a9 08 85  f5
cdf0 : 03 a9 01 85 02 a0 00 b1  41
cdf8 : 02 85 04 c8 b1 02 85 05  22
ce00 : c8 a5 14 91 02 c8 a5 15  f9
ce08 : 91 02 a5 14 18 65 fb 85  2e
ce10 : 14 a5 15 69 00 85 15 a5  35
ce18 : 04 85 02 a5 05 85 03 d0  3e
ce20 : d4 60 20 eb b7 86 a7 a5  43
ce28 : 14 85 a8 a5 15 85 a9 a9  55
ce30 : 3a 8d 02 03 a9 ce 8d 03  5f
ce38 : 03 60 a4 a8 a5 a9 20 95  fd
ce40 : b3 20 dd bd a2 00 bd 01  55
ce48 : 01 f0 06 9d 77 02 e8 d0  c3
ce50 : f5 a9 20 9d 77 02 e8 86  0e
ce58 : c6 20 60 a5 8a 48 98 48  d9
ce60 : a2 ff e8 bd 00 02 c9 30  8b
ce68 : 90 04 c9 3a 90 f4 bd 00  5c
ce70 : 02 e8 c9 20 f0 f8 c9 00  5b
ce78 : d0 0f a9 83 8d 02 03 a9  f3
ce80 : a4 8d 03 03 68 68 4c 83  0e
ce88 : a4 68 a8 68 aa 86 7a 84  69
ce90 : 7b 20 73 00 a2 ff 86 3a  b1
ce98 : 20 6b a9 48 8a 48 98 48  bf
cea0 : a5 a7 18 65 14 85 a8 a5  27
cea8 : 15 69 00 85 a9 68 a8 68  74
ceb0 : aa 68 4c 9f a4 20 9e b7  cb
ceb8 : a9 93 20 d2 ff e0 01 f0  7a
cec0 : 01 60 a2 14 4c c0 c5 20  3f
cec8 : eb b7 8a 0a 0a 0a 0a 18  bc
ced0 : 65 14 85 02 a9 00 8d ee  90
ced8 : ce a9 60 8d ef ce a9 00  61
cee0 : 85 fc a9 5c 85 fd a0 08  b4
cee8 : 88 c0 ff f0 23 b9 00 80  ef
cef0 : f0 f6 e6 fc d0 02 e6 fd  69
cef8 : ad ee ce 18 69 08 8d ee  be
cf00 : ce ad ef ce 69 00 8d ef  27
cf08 : ce c9 80 f0 12 4c e6 ce  b6
cf10 : a2 00 a5 02 81 fc e6 fc  f2
cf18 : d0 02 e6 fd 4c f8 ce 60  eb
cf20 : 20 9e b7 e0 01 f0 2f a9  41
cf28 : ea 8d e7 ca 8d e8 ca 8d  92
cf30 : 59 cb 8d 5a cb 8d ff cb  de
cf38 : 8d 00 cc 8d ca cb 8d cb  83
cf40 : cb a9 11 8d eb ca a9 ff  92
cf48 : 8d 97 cb a9 05 8d 01 cc  23
cf50 : a9 01 8d 5b cb 60 a9 49  42
cf58 : 8d e7 ca 8d 59 cb 8d ff  67
cf60 : cb 8d ca cb a9 ff 8d e8  c0
cf68 : ca 8d 5a cb 8d cb cb a9  c3
cf70 : 31 8d eb ca a9 00 8d 00  8d
cf78 : cc 8d 97 cb a9 25 8d 01  66
cf80 : cc a9 21 8d 5b cb 60 20  f1
cf88 : 00 98 20 d4 9a 4c b2 99  81
```

Listing 2. »Alan V7.2« (Schluß)

```
Name : alan ii                9000 9b91
---------------------------------------
9000 : a9 00 85 02 85 03 a9 01  64
9008 : 85 04 20 eb b7 20 4c 91  e6
9010 : a5 14 85 05 86 f8 20 fd  6e
9018 : ae 20 a5 cc 86 f9 a6 f7  9c
9020 : e4 15 90 28 f0 20 a5 05  69
9028 : 38 e5 14 85 fa a5 f7 e5  91
9030 : 15 85 fb a5 14 a6 15 85  91
9038 : 05 86 f7 a6 f9 a5 f8 85  0f
9040 : f9 86 f8 4c 5d 90 a6 14  61
9048 : e4 05 90 da a9 01 85 04  ef
9050 : a5 14 38 e5 05 85 fa a5  7e
9058 : 15 e5 f7 85 fb a6 f9 e4  b5
9060 : f8 90 0e a9 01 85 fc a5  d5
9068 : f9 38 e5 f8 85 b5 4c 7c  46
9070 : 90 a9 ff 85 fc a5 f8 38  d7
9078 : e5 f9 85 b5 a5 fb f0 03  76
9080 : 4c ac 90 a6 fa e4 b5 90  ea
9088 : 03 4c ac 90 a5 04 a6 fc  fe
9090 : 85 fc 86 04 a5 fa a6 b5  ee
9098 : 85 b5 86 fa a5 05 a6 f8  08
90a0 : 85 f8 86 05 a5 f7 85 b6  81
90a8 : a9 01 85 02 a9 00 85 b7  94
90b0 : 85 5a 85 5b 4c 0a 91 a5  d6
90b8 : 5b 18 65 b5 85 5b aa a5  58
90c0 : 5a 69 00 85 5a a8 c4 fb  76
90c8 : f0 2c 90 1a a5 5b 38 e5  18
90d0 : fa 85 5b a5 5a e5 fb 85  e8
90d8 : 5a a5 f8 18 65 fc 85 f8  8c
90e0 : a5 b6 69 00 85 b6 a5 05  e9
90e8 : 18 65 04 85 05 a5 f7 69  95
90f0 : 00 85 f7 4c 0a 91 e4 fa  f1
90f8 : 90 03 4c cc 90 a5 05 18  31
9100 : 65 04 85 05 a5 f7 69 00  29
9108 : 85 f7 a5 02 f0 10 a5 f8  4b
9110 : 85 14 a5 b6 85 15 a6 05  85
9118 : 20 ba ca 4c 2b 91 a5 05  b1
9120 : 85 14 a5 f7 85 15 a6 f8  a5
9128 : 20 ba ca a5 03 18 69 01  a5
9130 : 85 03 a5 b7 69 00 85 b7  b3
9138 : aa e4 fb 90 0b f0 03 4c  42
9140 : 4b 91 a4 fa c4 03 90 03  89
9148 : 4c b7 90 60 20 68 cc a5  64
9150 : 15 85 f7 60 20 d4 91 a9  74
9158 : 60 85 03 a9 00 85 02 a9  f8
9160 : 01 a2 04 a0 ff 20 ba ff  b3
9168 : a9 00 20 bd ff 20 c0 ff  d5
9170 : a2 01 20 c9 ff a0 00 b9  4d
9178 : d1 91 20 d2 ff c8 c0 03  c3
9180 : d0 f5 a2 19 a0 00 b9 cc  a1
9188 : 91 20 d2 ff c8 c0 05 d0  26
9190 : f5 a0 00 b1 02 20 d2 ff  78
9198 : c8 d0 f8 e6 03 a0 00 b1  7c
91a0 : 02 20 d2 ff c8 c0 40 d0  9c
91a8 : f6 c6 03 18 a5 02 69 40  56
91b0 : 85 02 a5 03 69 01 85 03  bb
91b8 : a9 0a 20 d2 ff ca d0 c4  ec
91c0 : 20 cc ff a9 01 20 c3 ff  9b
91c8 : 20 d4 91 60 1b 2a 00 40  46
91d0 : 01 1b 33 19 a9 00 a0 60  2d
91d8 : 8d e9 91 8d f7 91 8c ea  84
91e0 : 91 8c f8 91 a0 07 a2 07  02
91e8 : 5e 00 60 6a ca 10 f9 48  51
91f0 : 88 10 f3 a2 00 68 9d 00  8b
91f8 : 60 e8 e0 08 d0 f7 18 ad  8e
9200 : e9 91 69 08 8d e9 91 8d  97
9208 : f7 91 ad ea 91 69 00 8d  10
9210 : ea 91 8d f8 91 c9 80 d0  50
9218 : cb 60 ea ea ea ea ea ea  b3
9220 : ea ea ea ea ea ea ea ea  1f
9228 : ea ea ea ea ea ea ea ea  27
9230 : 20 eb b7 20 68 cc a5 14  e3
9238 : 85 fb a5 15 85 fc 86 fa  17
9240 : 20 fd ae 4c 5f 9b a5 14  26
9248 : 85 f9 86 f8 a9 00 8d 91  7e
9250 : 9b 8d 92 9b 8d 93 9b 8d  c9
9258 : 94 9b ee 91 9b ad 91 9b  4c
9260 : a8 b9 09 93 8d 68 92 20  40
9268 : 8d 93 ce 92 9b ee 94 9b  7f
9270 : ac 92 9b b9 0d 93 85 28  57
9278 : a5 f8 85 71 a9 00 85 29  2c
9280 : 85 72 20 57 b3 8a 8c 95  1e
9288 : 9b 4e 95 9b 6a 4e 95 9b  ca
9290 : 6a 4e 95 9b 6a 4e 95 9b  a1
9298 : 6a 4e 95 9b 6a 4e 95 9b  a9
92a0 : 6a 85 f7 a5 fa 38 e5 f7  78
92a8 : ac 94 9b 99 00 b2 a9 0c  0d
92b0 : 38 ed 92 9b a8 b9 0d 93  ab
92b8 : 85 28 a5 f9 85 71 a9 00  84
92c0 : 85 29 85 72 20 57 b3 8a  2a
92c8 : 8c 95 9b 4e 95 9b 6a 4e  4c
92d0 : 95 9b 6a 4e 95 9b 6a 4e  14
92d8 : 95 9b 6a 4e 95 9b 6a 4e  1c
92e0 : 95 9b 6a 85 f7 a5 fb 18  5b
92e8 : 65 f7 ac 94 9b 99 00 b0  ee
92f0 : a5 fc 69 00 ac 94 9b 99  7f
92f8 : 00 b1 ae 92 9b e0 01 f0  75
9300 : 03 4c 6a 92 4c 07 93 4c  fa
9308 : b9 93 20 49 6b 8d 01 0b  f9
9310 : 15 1e 27 2e 34 3a 3d 3f  4d
9318 : 40 3f 40 3a 3c 3e 3f 3f  80
9320 : a9 92 8d 06 93 a9 69 8d  7e
9328 : f2 92 a9 00 8d f3 92 a9  e4
9330 : 18 8d a5 92 a9 65 8d a6  14
9338 : 92 a9 5a 8d 05 93 a9 ee  59
9340 : 8d 6a 92 a9 0b 8d fe 92  1a
9348 : 60 a9 e9 8d f2 92 a9 00  13
9350 : 8d f3 92 a9 38 8d e7 92  65
9358 : a9 e5 8d e8 92 ee 92 9b  96
9360 : a9 ce 8d 6a 92 a9 01 8d  b7
9368 : fe 92 60 a9 e9 8d f2 92  f9
9370 : a9 00 8d f3 92 ce 92 9b  1c
9378 : a9 ee 8d 6a 92 a9 38 8d  bb
9380 : a5 92 a9 e5 8d a6 92 a9  41
9388 : 0b 8d fe 92 60 ee 92 9b  6b
9390 : a9 ce 8d 6a 92 a9 01 8d  e7
9398 : fe 92 a9 07 8d 05 93 a9  cd
93a0 : 93 8d 06 93 a9 18 8d e7  4f
93a8 : 92 a9 65 8d e8 92 a9 69  b6
93b0 : 8d f2 92 a9 00 8d f3 92  f1
93b8 : 60 a9 36 85 01 a0 00 c8  d2
```

Listing 3. »Alan II«, bitte mit dem MSE (Seite 159) eingeben.

```
93c0 : b9 00 b0 85 05 b9 00 b1   d8
93c8 : 85 f7 b9 00 b2 85 f8 c8   84
93d0 : b9 00 b0 85 14 b9 00 b1   d9
93d8 : 85 15 b9 00 b2 85 f9 84   9e
93e0 : 9e a9 00 85 02 a9 01 85   80
93e8 : 03 85 04 20 1e 90 a4 9e   e9
93f0 : c0 2c f0 03 4c c0 93 a0   bd
93f8 : 01 b9 00 b0 85 05 b9 00   53
9400 : b1 85 f7 b9 00 b2 85 f8   46
9408 : a0 2c b9 00 b0 85 14 b9   28
9410 : 00 b1 85 15 b9 00 b2 aa   a9
9418 : ca 86 f9 a9 00 85 02 a9   60
9420 : 01 85 03 85 04 20 1e 90   30
9428 : a9 37 85 01 60 ea ea ea   cd
9430 : 78 a9 45 8d 00 03 a9 95   6a
9438 : 8d 01 03 a9 01 85 bf 85   82
9440 : be 85 b7 8d b6 97 8d b2   24
9448 : 97 a9 00 8d b1 97 a9 6e   c1
9450 : 8d 14 03 a9 94 8d 15 03   ed
9458 : a9 db 8d 08 03 a9 95 8d   42
9460 : 09 03 a9 e4 8d 26 03 a9   5b
9468 : 95 8d 27 03 58 60 ad 03   33
9470 : 08 ae 04 08 85 b5 86 b6   5f
9478 : ad b1 97 f0 03 4c 31 ea   2f
9480 : a9 06 8d b1 97 4c 45 97   e6
9488 : ad b6 97 f0 08 20 ea e8   93
9490 : a9 00 8d b6 97 a9 01 85   49
9498 : b7 a9 e8 85 7a a9 07 85   2b
94a0 : 7b a5 7a 38 e9 28 85 7a   7e
94a8 : a5 7b e9 00 85 7b a0 00   3c
94b0 : b1 7a a2 30 86 02 c5 02   e1
94b8 : f0 26 e6 02 a6 02 e0 3a   28
94c0 : f0 03 4c b6 94 a5 7b c9   14
94c8 : 04 d0 d6 a5 7a c9 00 f0   76
94d0 : 03 4c a1 94 a9 00 8d b1   28
94d8 : 97 a9 01 85 be 4c 31 ea   1e
94e0 : 99 f8 96 c8 b1 7a c0 05   b0
94e8 : d0 f6 a9 94 a0 fa 8d ff   48
94f0 : 01 8c fe 01 a2 f9 9a 4c   14
94f8 : 31 ea a5 bf f0 0e a2 f7   fa
9500 : a0 96 86 7a 84 7b 20 73   68
9508 : 00 20 6b a9 a9 00 85 d3   81
9510 : a9 17 85 d6 20 10 e5 20   db
9518 : 13 a6 a0 00 b1 5f aa c8   f9
9520 : b1 5f 85 60 86 5f b1 5f   d7
9528 : d0 0a a9 00 8d b1 97 85   37
9530 : be 4c 86 e3 c8 b1 5f 85   d5
9538 : 14 c8 b1 5f 85 15 a9 00   b0
9540 : 85 bf 4c c9 a6 ad b2 97   c3
9548 : d0 2d ad b7 97 c9 2a 90   a3
9550 : 0a 20 5c 97 20 78 97 a9   ec
9558 : 04 85 da 20 5c 97 20 78   cd
9560 : 97 a9 84 85 da a9 01 8d   b8
9568 : b2 97 ad b4 97 ae b5 97   dd
9570 : 85 5f 86 60 4c c9 a6 ad   5b
9578 : b1 97 d0 03 4c 8b e3 a9   8d
9580 : 00 85 cc a5 14 c5 b5 d0   12
9588 : 0d a5 15 c5 b6 d0 07 a9   c7
9590 : 00 85 b7 4c 9e 95 a0 01   e5
9598 : b1 5f d0 02 85 be 20 66   09
95a0 : e5 a9 00 8d b1 97 6c 02   99
95a8 : 03 a5 cb c9 04 f0 0c c9   35
95b0 : 05 f0 16 a9 00 8d b1 97   4a
95b8 : 4c 31 ea a5 be f0 f4 a9   a7
95c0 : 01 85 cc 8d b8 97 4c 88   f3
95c8 : 94 a5 b7 f0 e6 a9 01 85   05
95d0 : cc 8d b6 97 85 bf 85 be   ed
95d8 : 4c fa 95 a9 01 85 bf 85   82
95e0 : be 4c 3d c2 8d b3 97 ad   9c
95e8 : b2 97 d0 08 ee b7 97 a9   f9
95f0 : 13 4c ca f1 ad b3 97 4c   89
95f8 : ca f1 ad b8 97 f0 11 a9   d6
9600 : 00 8d b8 97 20 5c 97 20   6b
9608 : 78 97 a9 84 85 d9 85 da   3a
9610 : a2 00 a9 20 9d 00 04 e8   dc
9618 : e0 28 d0 f8 a9 00 85 7a   05
9620 : a9 04 85 7b a5 7a 18 69   fd
9628 : 28 85 7a a5 7b 69 00 85   74
9630 : 7b a0 00 b1 7a a2 30 86   bc
9638 : 02 c5 02 f0 26 e6 02 a6   aa
9640 : 02 e0 3a f0 03 4c 39 96   04
9648 : a5 7b c9 07 d0 d6 a5 7a   4d
9650 : c9 c0 f0 03 4c 24 96 a9   a9
9658 : 00 8d b1 97 a9 01 85 b7   a6
9660 : 4c 31 ea 99 f8 96 c8 b1   fd
9668 : 7a c0 05 d0 f6 a9 96 a0   f6
9670 : 7d 8d ff 01 8c fe 01 a2   de
9678 : f9 9a 4c 31 ea a9 00 85   ff
9680 : d3 85 d6 20 10 e5 a2 f7   7a
9688 : a0 96 86 7a 84 7b 20 73   f0
9690 : 00 20 6b a9 20 13 a6 a5   31
9698 : 5f 85 fb a5 60 85 fc a5   df
96a0 : 5f 38 e9 04 85 5f a5 60   c1
96a8 : e9 00 85 60 a5 5f 38 e9   09
96b0 : 01 85 5f 8d b4 97 a5 60   5d
96b8 : e9 00 85 60 8d b5 97 a0   35
96c0 : 00 b1 5f c9 00 d0 e5 c8   5a
96c8 : b1 5f c5 fb d0 de c8 b1   a4
96d0 : 5f c5 fc d0 d7 a5 60 c9   2b
96d8 : 08 b0 0a a9 00 8d b1 97   52
96e0 : 85 b7 4c 86 e3 4c 8a 97   1f
96e8 : a9 00 8d b7 97 8d b2 97   cb
96f0 : 4c c9 a6 b7 4d b7 4d b7   f9
96f8 : 32 34 30 20 01 b7 4d b7   c7
9700 : 20 9e b7 e0 00 f0 03 4c   a5
9708 : 30 94 a2 00 bd 2e 97 20   17
9710 : d2 ff e8 e0 17 d0 f5 78   f9
9718 : a9 8b 8d 00 03 a9 e3 8d   12
9720 : 01 03 a9 31 8d 14 03 a9   0c
9728 : ea 8d 15 03 58 60 4c 49   cb
9730 : 53 54 2d 53 43 52 4f 4c   00
9738 : 4c 20 44 45 41 4b 54 49   a0
9740 : 56 49 45 52 54 a5 cb c9   0c
9748 : 06 f0 03 4c a9 95 a9 00   ff
9750 : a2 18 85 d3 86 d6 20 10   9a
9758 : e5 4c b3 95 a9 00 a2 04   30
9760 : 85 5f 86 60 a9 bf a2 07   74
9768 : 85 5a 86 5b a9 e7 a2 07   9a
9770 : 85 58 86 59 20 bf a3 60   3d
9778 : a2 18 b5 d9 29 7f b4 d8   e2
9780 : 10 02 09 80 95 d9 ca d0   d9
9788 : f1 60 c8 b1 5f 85 14 c8   16
9790 : b1 5f 85 15 a9 00 8d b7   35
9798 : 97 8d b2 97 ee b4 97 e6   56
97a0 : 5f d0 05 ee b5 97 e6 60   fb
97a8 : 4c c9 a6 ea ea ea ea ea   67
97b0 : ea ea ea ea ea ea ea ea   af
97b8 : ea ea ea ea ea ea ea ea   b7
97c0 : ea ea ea ea ea ea ea ea   bf
97c8 : ea ea ea ea ea ea ea ea   c7
97d0 : ea ea ea ea ea ea ea ea   cf
97d8 : ea ea ea ea ea ea ea ea   d7
97e0 : ea ea ea ea ea 50 00 00   88
97e8 : 02 ea 02 ea 0a ea 0a ea   33
97f0 : 52 ea c2 00 04 ea d8 ea   39
97f8 : a0 ea a4 ea 00 0a 02 02   f0
9800 : a9 00 8d e6 97 8d e7 97   9e
9808 : ad f3 97 c9 02 f0 03 4c   1a
9810 : 1d 98 20 e0 99 20 e0 99   ef
9818 : 4c 62 99 ea ea ea ea 20   4b
9820 : 8f 99 20 eb b7 a5 14 8d   15
9828 : ff 97 8e fe 97 20 fd ae   46
9830 : 20 73 99 a5 14 8d fd 97   fa
9838 : 8e fc 97 20 fd ae 20 9e   41
9840 : b7 4c 46 99 ea a0 00 a9   e9
9848 : a0 8d df 98 a9 a4 8d e8   81
9850 : 98 4c 60 98 a0 01 a9 a8   43
9858 : 8d df 98 a9 ac 8d e8 98   3c
9860 : 20 55 99 ad fe 97 99 e8   2c
9868 : 97 ad ff 97 99 ea 97 ad   73
9870 : fd 97 99 ee 97 ad fc 97   87
9878 : 99 ec 97 ad fe 97 0a 2e   54
9880 : e7 97 0a 2e e7 97 0a 2e   3b
9888 : e7 97 0a 2e e7 97 0a 2e   43
9890 : e7 97 8d e5 97 ad fe 97   75
9898 : 0a 2e e6 97 0a 2e e6 97   43
98a0 : 0a 2e e6 97 18 6d e5 97   22
98a8 : 8d e5 97 a9 00 69 00 85   99
98b0 : f7 ad e5 97 6d ff 97 99   52
98b8 : f0 97 85 fe 85 fc 85 fa   01
98c0 : 85 f8 ad e6 97 65 f7 6d   69
98c8 : e7 97 aa 18 69 04 99 f4   2f
98d0 : 97 85 ff 8a 18 69 d8 99   de
98d8 : f6 97 85 fd 8a 18 69 a0   0b
98e0 : 99 f8 97 85 fb 8a 18 69   d3
98e8 : a4 99 fa 97 85 f9 ac fd   e1
98f0 : 97 a2 00 88 b1 fe 91 fa   39
98f8 : b1 fc 91 f8 a9 a0 91 fe   8f
9900 : a9 0e 91 fc c0 00 d0 eb   db
9908 : a5 fe 18 69 28 85 fe 85   15
9910 : fc 85 fa 85 f8 a9 00 69   ee
9918 : 00 85 f7 a5 fd 65 f7 85   83
9920 : fd a5 ff 65 f7 85 ff a5   93
9928 : f9 65 f7 85 f9 a5 fb 65   0a
9930 : f7 85 fb ce fc 97 d0 b6   00
9938 : 18 ae fe 97 ac ff 97 20   c3
9940 : 0a e5 ee f3 97 60 e0 02   7b
9948 : f0 05 b0 06 4c 45 98 4c   92
9950 : 54 98 4c 62 99 98 cd f2   cb
9958 : 97 f0 04 8c f2 97 60 4c   00
9960 : 15 98 20 53 e4 20 bf e3   4a
9968 : ea ea ea 20 00 c0 a2 0e   33
9970 : 4c 37 a4 20 eb b7 ad ff   b8
9978 : 97 18 65 14 c9 29 b0 0a   b4
9980 : 8a 18 6d fe 97 c9 1a b0   e3
9988 : 01 60 a2 0e 4c 3a a4 a9   a0
9990 : 9a 8d 18 03 a9 99 8d 19   27
9998 : 03 60 a9 00 8d f3 97 a9   60
99a0 : c1 8d f2 97 a9 3d 8d 08   a2
99a8 : 03 a9 c2 8d 09 03 4c 47   4b
99b0 : fe 40 78 a9 bf 8d 08 03   b0
99b8 : a9 99 8d 09 03 58 60 a9   7a
99c0 : 00 8d b1 99 a0 00 ea ea   b2
99c8 : ea 88 d0 fa ad b1 99 38   c9
99d0 : e9 20 8d b1 99 aa e0 40   56
99d8 : f0 03 4c c4 99 4c 3d c2   6c
99e0 : ad f3 97 c9 00 f0 03 4c   d2
99e8 : ed 99 4c 08 af a9 36 85   e2
99f0 : 01 ac f2 97 b9 f0 97 85   83
99f8 : f8 85 fa 85 fc 85 fe b9   8d
9a00 : f4 97 85 fb b9 ea 97 8d   0d
9a08 : ff 97 b9 e8 97 8d fe 97   6f
9a10 : b9 f6 97 85 f9 b9 f8 97   5b
9a18 : 85 ff b9 fa 97 85 fd b9   7b
9a20 : ee 97 8d fd 97 b9 ec 97   27
9a28 : 8d fc 97 ac fd 97 a2 00   d6
9a30 : 88 b1 fe 91 fa b1 fc 91   d7
9a38 : f8 c0 00 d0 f3 a5 fe 18   43
9a40 : 69 28 85 fe 85 fc 85 fa   4b
9a48 : 85 f8 a9 00 69 00 85 f7   50
9a50 : a5 ff 65 f7 85 ff a5 fd   38
9a58 : 65 f7 85 fd a5 fb 65 f7   9a
9a60 : 85 fb a5 f9 65 f7 85 f9   ab
9a68 : ce fc 97 d0 be a9 37 85   d5
9a70 : 01 ad f3 97 c9 02 f0 03   ae
9a78 : 4c 94 9a ad f2 97 49 01   7e
9a80 : a8 8d f2 97 b9 ea 97 85   fb
9a88 : d3 b9 e8 97 85 d6 20 10   15
9a90 : e5 4c b2 9a 78 a9 00 8d   8b
9a98 : 1a d0 a9 31 8d 14 03 a9   84
9aa0 : ea 8d 15 03 a9 3d 8d 08   c1
9aa8 : 03 a9 c2 8d 09 03 8d f2   a7
9ab0 : 97 58 ce f3 97 60 20 eb   7a
9ab8 : b7 a5 fb 18 65 14 c9 29   b4
9ac0 : b0 09 8a 18 65 fd c9 1a   3c
9ac8 : b0 01 60 a2 0e 4c 3a a4   db
9ad0 : 0a 0b 03 02 78 a9 e6 8d   ec
9ad8 : 14 03 a9 9a 8d 15 03 a9   0c
9ae0 : 81 8d 1a d0 58 60 ad 19   3a
9ae8 : d0 8d 19 d0 30 07 ad 0d   eb
9af0 : dc 58 4c 31 ea 38 20 0a   37
9af8 : e5 8e d2 9a 8c d3 9a ac   57
9b00 : f2 97 b9 ea 97 18 79 ee   87
9b08 : 97 38 e9 01 8d d1 9a ad   83
9b10 : d3 9a cd d1 9a 10 25 b9   10
9b18 : e8 97 18 79 ec 97 38 e9   41
9b20 : 02 8d d0 9a ec d0 9a 10   50
9b28 : 1b ad d3 9a d9 ea 97 30   16
9b30 : 1b ad d2 9a d9 e8 97 30   cd
9b38 : 1b 4c 81 ea ad d1 9a 85   16
9b40 : d3 4c 17 9b ad d0 9a 85   49
9b48 : d6 4c 29 9b b9 ea 97 85   5e
9b50 : d3 4c 81 ea b9 e8 97 85   53
9b58 : d6 20 10 e5 4c 81 ea 20   bc
9b60 : eb b7 a5 14 85 f9 86 f8   47
9b68 : 18 65 fb 85 14 a5 15 65   70
9b70 : fc 85 15 8a 18 65 fa aa   b3
9b78 : 20 68 cc a5 fc d0 06 a5   6e
9b80 : fb c5 f9 90 09 a5 fa c5   23
9b88 : f8 90 03 4c 4c 92 4c 48   2e
9b90 : b2 fe 00 ff 00 ff 00 ff   c1
```

Listing 3. (Schluß)

```
10 HGR 1:GCLR:SET 1,0:CCOPY
20 FOR I=1 TO 90 STEP 5
30 REC 160-I,100-I,I,I
40 NEXT I
50 FOR I=1 TO 90 STEP 5
60 REC 160,100,I,I
```

Listing 4. Dieses Demoprogramm erzeugt die Bilder 1 bis 5. Bitte geben Sie es nur bei aktivem »Alan« ein.

Zwei tolle 64'er Sonderhefte mit vielen Tips & Tricks & Tools

Das 64'er Sonderheft 0002
Grundlagen. Debugging – Fehlersuche in Basic-Programmen. Grafik: Super-Hardcopy bringt jeden Bildschirminhalt auf einen MPS 802.
Tips & Tricks-Listings: Datasette schneller als Floppy durch Tornado-Tape / Flottes kopieren mit »Express-Copy« / Filemanager ordnet Disketten / POKEs die man kennen sollte / 3D-Grafik für Schachspiele / Interessante Tips & Tricks und Einzeiler aus 64'er.

SONDERHEFT 2/86
Markt&Technik
64'er
TIPS & TRICKS SONDERHEFT
TIPS & TRICKS
Super Listings
Ausführliche Grundlagen
★ Floppy
★ Datasette
★ Drucker
★ Peeks & Pokes
★ Einzeiler
★ Sprites
★ Grafik
★ Musik
★ Basic
★ Dateiverwaltung
★ Fehlersuche
★ Assembler
Alle Programme auf Diskette erhältlich



Das 64'er Sonderheft 9901
Viele Listings für flottes und komfortables Programmieren.
Checksummer: keine Fehler mehr beim Abtippen von Listings / M-P-S: mehr als 30 Programme gleichzeitig im Speicher / ESF: Editieren sequentieller Dateien / Track 18: Das Chaos organisieren / P-Basic-V2: Autostart mit Rückwärtsgang / Eigene Basic-Befehle auf dem C 64 / Super Line – 80 Zeichen für den C 64.

Nutzen Sie die Bestellmöglichkeiten für diese zwei »64'er«-Sonderhefte mit der eingehefteten Zahlkarte in diesem Sonderheft oder fragen Sie Ihren Zeitschriftenhändler.

```
70 NEXT I
80 FOR I=1 TO 50 STEP 5
90 CIRCLE 160+I,100-I,I,I
100 NEXT I
110 FOR I=1 TO 50 STEP 5
120 CIRCLE 160-I,100+I,I,I
130 NEXT I
155 WAIT 198,255
156 GCLR
157 FOR I=10 TO 90 STEP 10
158 REC 160-I,100-I,2*I,2*I
159 NEXT I
160 FOR I=0 TO 100 STEP 5
170 REC 269-I,159-I,50,40
180 NEXT I:BLOCK 269-I,159-I,50,40
190 FOR I=0 TO 100 STEP 5
200 REC 0+I,0+I,50,40
210 NEXT I:BLOCK 0+I,0+I,50,40
220 POKE 198,0:WAIT 198,255
230 GCLR
240 FOR I=1 TO 50 STEP 4
250 CIRCLE 160-I,100-I,ABS(INT(SIN(I)*40)),I
260 NEXT I
270 FOR I=1 TO 50 STEP 4
280 CIRCLE 160+I,100+I,ABS(INT(SIN(I)*40)),I
290 NEXT I
300 FOR I=1 TO 50 STEP 7
310 LINE 160,100,190+ABS(INT(COS(I)*40)),I
320 NEXT I
330 FOR I=1 TO 50 STEP 7
340 LINE 160,100,ABS(INT(COS(I)*40)),130+I
350 NEXT I
360 POKE 198,0:WAIT 198,255
370 GCLR
380 FOR I=0 TO 319
390 VLINE I,0,100+75*SIN(π*4*I/278)
400 NEXT I
405 POKE198,0:WAIT198,255
410 HGR1:GCLR:GCLR:R=50
420 FOR X=1 TO 319 STEP 3
430 LINEX,40,50*SIN(X/30)+100,X/1.6
440 NEXT X
450 POKE198,0:WAIT198,255:GCLR
455 FOR X=1 TO 162 STEP 3
460 CIRCLE160,100,30*SIN(X/10)+100,X/1.6
470 NEXT X
1000 POKE198,0:WAIT 198,255
1010 TEXT:HOME1:VTAB6:HELP:END
```

Listing 4. (Schluß)

Zeichen flink erstellt

Mit diesem Programm können Sie einfach und schnell Ihren eigenen, ganz persönlichen Zeichensatz entwerfen. Als Bonbon enthält das Programm eine Routine, mit der sich Zeichensätze auf Diskette generieren lassen, die wie ganz normale Basic-Programme geladen und gestartet werden.

Der »Character-Editor II« (Listing 1) ist ein Programm, mit dem sich leicht und komfortabel eigene Zeichensätze erstellen lassen. Nach dem Laden mit »LOAD "CHR-EDITOR II",8« wird das Programm durch den Befehl RUN gestartet. Anschließend stehen dem Benutzer die beiden Zeichensätze des C 64 zur Bearbeitung zur Verfügung. Der Editor verfügt über 21 Routinen, die das Erstellen eines neuen Zeichensatzes in sehr komfortabler Weise unterstützen. Die Routinen sind in einer Menütafel innerhalb des Arbeitsbereiches durch die Cursortasten frei anwählbar. Der Character-Editor ist dabei sehr übersichtlich gehalten. Es wird immer das gerade zu bearbeitende Zeichen, dessen Code, die entsprechende 8x8-Matrix und der aktuelle Zeichensatz verarbeitet (Bild 1). Das Hin- und Herspringen zwischen verschiedenen Menüs entfällt. Alle Ein- und Ausgaben werden in einer speziellen Zeile im unteren Bereich des Bildschirms verarbeitet. Die Steuerung innerhalb der Routinen wird ebenfalls mit den Cursortasten erledigt, so daß der Editor mit wenigen Tasten schnell und einfach zu bedienen ist. Weiterhin verfügt der Zeichensatz-Editor über einen Testmodus, in welchem der Zeichensatz in der oberen Bildschirmhälfte eingeblendet ist, und so die gesamte Tastaturbelegung ausprobiert werden kann.

Die bearbeiteten Zeichensätze lassen sich auf Diskette speichern. Der fertig generierte Zeichensatz kann aber auch als eigenständiges Programm mit Basic-Start gespeichert werden. Das bedeutet, daß ein geänderter Zeichensatz mit »LOAD "ZEICHENSATZ",8« geladen und mit RUN gestartet wird.

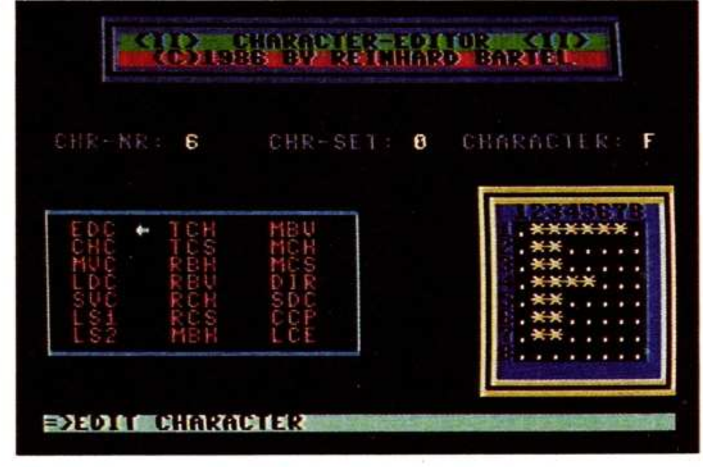


Bild 1. Das Arbeitsfeld des Zeichensatz-Editors

Alles in allem steht Ihnen ein sehr leistungsfähiges Werkzeug zur Verfügung, mit dem ein den individuellen Bedürfnissen und Wünschen angepaßter Zeichensatz erstellt werden kann.

Das Zeichen, welches editiert werden soll, kann jederzeit direkt über die Tastatur eingegeben werden (normal oder in Kombination mit der SHIFT- oder CBM-Taste). Das Setzen oder Löschen eines Punktes erfolgt mit der CBM-Taste.

Es stehen aber noch weitere, über die Tastatur zu erreichende Funktionen zur Verfügung. Diese sind:

<INS> – Anwahl des zu bearbeitenden Zeichens durch Erhöhen des Character-Codes um den Wert 1.

<DEL> – Anwahl des zu bearbeitenden Zeichens durch Erniedrigen des Character-Codes um den Wert 1.

<HOME> – Anwahl des zu bearbeitenden Zeichens durch Erhöhen beziehungsweise Erniedrigen des Character-Codes um den Wert 128.

<F1> – Austauschen der beiden zu bearbeitenden Zeichensätze.

<F2> - Kopiert die Zeichen der Character-Codes 0 bis 127 als die Zeichen der Character-Codes 128 bis 255 und invertiert diese.

<F3> - Invertieren des aktuellen Zeichens.

<F4> - Löscht das aktuelle Zeichen.

<F5> - Sprung in den Testmodus und zurück mit <RETURN>.

<F6> - Schaltet den editierten Zeichensatz ein beziehungsweise aus. Die Meldung »Zeichensatz nicht aktiviert« wird entsprechend ein- oder ausgeblendet.

<F7> - Speichert das aktuelle Zeichen.

<F8> - Übernimmt das gespeicherte Zeichen anstelle des aktuellen Zeichens.

Diese Funktionen sind in jeder Bearbeitungsroutine jederzeit erreichbar.

Anwahl der Bearbeitungsroutinen

Die Abkürzungen der Bearbeitungsroutinen befinden sich in einer kleinen Menütafel innerhalb des Bildschirms. Mit den Cursortasten kann ein Pfeil gesteuert werden, der immer auf die momentan angewählte Routine zeigt. Gleichzeitig wird der volle Text der angewählten Funktion in einer speziellen Zeile im unteren Teil des Bildschirms ausgegeben. Ein Druck auf die RETURN-Taste aktiviert die gewünschte Funktion. Ein weiterer Druck auf <RETURN> verläßt die Funktion wieder. Der Pfeil erscheint dann wieder und die reverse Darstellung der aktiven Funktion erlischt. Doch nun die Routinen im einzelnen:

EDC (Edit Character)

Editieren eines Zeichens. Es erscheint der Cursor innerhalb der 8x8-Matrix des aktuellen Zeichens. Mit den Cursortasten kann nun jedes Bit des Zeichens angefahren werden. Ein Druck auf die CBM-Taste löscht oder setzt den betreffenden Punkt.

CHC (Change Characters)

Vertauschen zweier Zeichen. Mit den Cursortasten kann das erste zu vertauschende Zeichen angewählt werden. Ist dies erledigt, ist die Aufforderung »Select 1. Character« mit <RETURN> zu bestätigen. Ebenso ist mit der Anwahl des zweiten Zeichens zu verfahren. Wurde die Aufforderung »Select 2. Character« mit <RETURN> quittiert, erscheint die Sicherheitsabfrage »Are you sure (Y/N)?«. Bestätigen Sie diese Frage mit <Y>, tauscht das Programm die beiden Zeichen miteinander aus. Es lassen sich Zeichen aus dem ersten Zeichensatz mit Zeichen aus dem zweiten Zeichensatz austauschen, wenn mit <F1> der Zeichensatz umgeschaltet wird.

MVC (Move Character)

Verschieben (Kopieren) eines Zeichens beziehungsweise Auswahl dieses Zeichens. Wie sich ein Zeichen verschieben läßt, wurde bereits bei der Routine »CHC« beschrieben.

LDC (Load Character-Set)

Laden eines zuvor mit der SVC-Routine gespeicherten Zeichensatzes. Der Name des zu ladenden Zeichensatzes wird in der untersten Zeile eingegeben.

SVC (Save Character-Set)

Speichern der beiden in Bearbeitung befindlichen Zeichensätze. Eingabe des Namens wie bei der LDC-Routine.

LS1 (Load Set 1)

Laden des ersten C 64-Zeichensatzes. Wird die Sicherheitsabfrage mit <Y> bestätigt, wird der erste der beiden originalen Zeichensätze als aktueller Zeichensatz übernommen. Der gerade bearbeitete Zeichensatz geht dabei verloren.

LS2 (Load Set 2)

Laden des zweiten C 64-Zeichensatzes. Wie LS1-Routine, nur wird der zweite Zeichensatz geladen.

TCH (Turn Character)

Drehen eines Zeichens. Mit den Cursortasten kann das aktuelle Zeichen nach rechts und links gedreht werden.

TCS (Turn Character-Set)

Drehen sämtlicher Zeichen des aktuellen Zeichensatzes mit Hilfe der Cursortasten.

RBH (Rotate Byte Horizontal)

Horizontales Rotieren eines Bytes innerhalb eines Zeichens. Mit den Tasten <CURSOR-abwärts/-aufwärts> wird ein Pfeil innerhalb der 8x8-Matrix gesteuert, der das zu rotierende Byte bestimmt. Mit den Tasten <CURSOR-rechts/-links> wird das selektierte Byte entsprechend rotiert.

RBV (Rotate Byte Vertikal)

Vertikales Rotieren eines Bytes innerhalb eines Zeichens. Mit den Tasten <CURSOR rechts/-links> wird ein Pfeil innerhalb der 8x8-Matrix gesteuert, der das zu rotierende Byte bestimmt. Die Tasten <CURSOR aufwärts/-abwärts> lassen das angegebene Byte entsprechend rotieren.

RCH (Rotate Character)

Rotieren eines Zeichens. Mit den Cursortasten kann das aktuelle Zeichen in jede Richtung byteweise rotiert werden.

RCS (Rotate Character-Set)

Rotieren sämtlicher Zeichen des in Bearbeitung stehenden Zeichensatzes mit Hilfe der Cursortasten.

MBH (Mirror Byte Horizontal)

Horizontales Spiegeln eines Bytes innerhalb eines Zeichens. Die Steuerung erfolgt wie bei der RBH-Routine.

MBV (Mirror Byte Vertikal)

Vertikales Spiegeln eines Bytes innerhalb eines Zeichens. Die Steuerung erfolgt wie bei der RBH-Routine.

MCH (Mirror Character)

Spiegeln eines Zeichens. Mit den Cursortasten kann das aktuelle Zeichen horizontal und vertikal gespiegelt werden.

MCS (Mirror Character-Set)

Spiegeln des aktuellen Zeichensatzes. Mit den Cursortasten kann der aktuelle Zeichensatz horizontal und vertikal gespiegelt werden.

DIR (Directory)

Anzeigen des Directory der eingelegten Diskette.

SDC (Send Disk-Command)

Senden von Diskettenbefehlen. In der Eingabezeile im unteren Teil des Bildschirms kann jeder Befehl an die Diskettenstation eingegeben werden (etwa SCRATCH, VALIDATE). Nach beendeter Diskettenoperation wird der Status des Laufwerks angezeigt.

CCP (Create Character-Program)

Erzeugen eines Zeichensatz-Programms. Diese Funktion generiert einen Zeichensatz mit Basic-Start. Der momentan bearbeitete Zeichensatz wird zusammen mit einer Aktivierungsroutine auf Diskette gespeichert. Die Eingabe des Namens erfolgt in der untersten Bildschirmzeile. Vor den Namen wird automatisch »C/« als Kennung für ein Zeichensatz-Programm gestellt.

LCE (Leave Character-Editor)

Verlassen des Character-Editors. Nach Bestätigen der Sicherheitsabfrage mit <Y> befinden Sie sich wieder im Direktmodus des C 64. Soll der zuletzt bearbeitete Zeichensatz weiter editiert werden, so ist der Zeichensatz-Editor mit »PRINT USR(0)« zu starten. Ansonsten kann mit RUN ganz neu begonnen werden.

Allgemeine Hinweise

- Der Zeichensatz-Editor ist vollständig in Maschinensprache geschrieben und befindet sich im Speicher von $0801 bis $1D82.
- Es wird immer nur ein Zeichensatz editiert (256 Zeichen). Man kann aber zwischen den Zeichensätzen frei umschalten.



Software der Extra-Klasse

Ab sofort kommen alle Besitzer eines C64 und C128 in den Genuß unschlagbarer Software. Mit unserer neuen Produkt-Reihe 64'er-EXTRA erhalten Sie brandaktuelle Top-Programme zu den beliebtesten Themen: Super-Spiele, Tools, Anwendungen sowie Best-of-Pakete. Mit jedem Software-Paket wird ein Bedienungshandbuch für alle Programme mitgeliefert. Sie erhalten so das beste und interessanteste Werkzeug für Ihren Computer und können es sofort einsetzen.



Best of Grafik Vol. 1

Giga-CAD Unschlagbare 3D-Konstruktion auf dem C 64 · ***Hi-Eddi*** Das Super-Zeichen- und Malprogramm · ***Title Wizard*** Giga-CAD-Filme für eigene Vorspänne · ***Pic-Loader*** Verwenden Sie Hi-Eddi-Grafiken für eigene Programme · ***Hi-Maus*** Maus-Treiber für Hi-Eddi · ***Hi-Spiegel*** Spiegeln Sie beliebige Ausschnitte einer Grafik · ***Filmconverter*** Giga-Cad-Filme können mit diesem Programm in das Hi-Eddi-Format umgewandelt werden · ***Druckeranpassungen für Hi-Eddi:*** Printer/Plotter VC 1520, MPS-801/802/803, Seikosha GP 700VC, Star NL-10, Commodore-Plotter VC 1520, C.Itoh-8510, C.Itoh Riteman C+

1 Programmdiskette für C 64/ C 128
1 Demonstrationsdiskette für Giga-CAD
Bestell-Nr. 38701 sFr 44,90*/öS 499,-* **DM 49,90***



Best of Grafik Vol. 2

Grafik 2000 43 neue Grafikbefehle für Ihren C 64 · ***Provic 64*** 32 Sprites gleichzeitig auf dem Bildschirm · ***Grafic-Calc*** Professionelle Business-Grafik – vom Säulen- zum Kuchendiagramm · ***3D-Grafik-Master*** Drehen dreidimensionaler Körper in Echtzeit · ***MPS-Support*** Profi-Auflösung für MPS-Drucker · ***Epson-Support*** Grafik-Befehlserweiterung für Ihren Drucker · ***Scroll-Machine*** Ruckfreie Scroll-Routine zum Einbinden in eigene Spiele · ***Pseudo-Scroll*** Eine Routine, mit der Sie Bewegung ins Spiel bringen · ***Fractal-Berge*** Bizarre Landschaften aus dem Computer · ***Grafik-Wandler*** Rechnet Hires-Grafiken in Lores-Bilder um · ***Lores zu Hires*** Transportiert Textbildschirme in hochauflösende Grafik · ***Pic-Loader*** Grafikdieb für Printshop und Printmaster · ***Hardmaker*** Raubt und druckt Zeichensätze und Hires-Grafiken aus professionellen Programmen · ***Hardcopy-Routinen*** Super-Treiber für MPS 801, Epson, VC 1520 und CP-80X

1 Diskette für C 64/ C 128
Bestell-Nr. 38702 sFr 34,90*/öS 399,-* **DM 39,90***



Best of Grafik Vol. 3

Hires-Master Die wohl schnellste Grafikerweiterung · ***Sprite + Grafik-Basic*** Mehr als 100 neue Befehle für Ihren C 64 · ***3D-Schach-Grafik*** Tolle Befehle zur Schachprogrammierung · ***IRQ-Basic*** Grafik und Musik im Interrupt · ***Game-Basic*** Neue Befehle zur Programmierung von Spielen · ***Kudi 64*** Mathematische Funktionen grafisch dargestellt · ***Shapes 64*** Bringen Sie Bewegung ins Bild · ***Apfelmännchen*** Bilder aus einer anderen Dimension · ***3D-Movie-Maker*** Trickfilme in der vierten Dimension · ***Chartset-Master*** Zeichensatzgenerator mit über 100 Befehlen · ***Graphic-Art*** Editor als Antwort auf das Sprite-Problem · ***Super-Hardcopy*** Unschlagbare Druckqualität für Epson-Drucker · ***Epson-Plotter*** Matrixdrucker simuliert Plotter · ***Hardcopy-Programme*** für Epson-Drucker, Star SG-10, MPS 801/802/803

1 Diskette für C 64/ C 128
Bestell-Nr. 38703 sFr 34,90*/öS 399,-* **DM 39,90***

Jetzt lieferbar:

Adventure-Pack Vol. 1

Robox
Fesselndes Grafik-Science-Fiction-Adventure
Der Herrscher eines fremden Planeten ließ sein Gehirn nach seinem Tod künstlich weiterleben – in einem Körper ohne Seele. Aus dieser Kombination, halb Roboter und halb Mensch, halb lebend und halb tot, die Robox genannt wurde, enstand der Haß auf alles Lebende. Er befahl daher seiner Roboter-Armee, sämtliches Leben des Universums zu vernichten. Ihre Aufgabe ist es, zu Robox zu gelangen und ihn unschädlich zu machen, um die Erde vor ihrem sonst sicheren Tod zu bewahren. Wie Sie dies tun, bleibt Ihnen überlassen.

Scotland Yard
Spannendes Kriminal-Adventure
Begeben Sie sich auf spannende Verbrecherjagd in das London des 19. Jahrhunderts, und lassen Sie sich engagieren bei Scotland Yard. Verhören Sie Tatverdächtige, prüfen Sie deren Alibis und verfolgen Sie die Spuren zurück zum Täter. Als Belohnung für die Lösung der zehn ungeklärten Fälle wartet Ihre Beförderung zum Oberinspektor.
Mit dem mitgelieferten Fall-Editor konstruieren Sie weitere Verbrechen und geben damit Ihren Freunden harte Nüsse zu knacken.

3 Disketten (beidseitig bespielt) für den C 64/ C 128
Bestell-Nr. 38704 sFr 24,90*/öS 299,-* **DM 29,90***

*Unverbindliche Preisempfehlung

The Best of Floppy Tools Volume 1

Lieferumfang: Anleitungsheft · 5¼"-Diskette (1541-Format)
Disc-Wizard Directory-Sort, Diskmonitor, Text-Search u.v.m. · ***Copy +*** Parallel-Backup-Kopierprogramm · ***Super-Copy*** Komfortables File-Kopierprogramm · ***Thirty Seconds*** Blitzschnelles Backup-Kopierprogramm für C128 · ***FMON 1541*** leistungsfähiger Floppy-Monitor · ***Disk-Monitor*** kurzer Disk-Monitor für C64 · ***Ultraload/-boot Plus*** Software-Speeder und Autostart-Generator für Schnellader · ***EX-SMON-DOS*** brennfähige EPROM-Version des Betriebssystems · ***Datawork Basic*** Komfortable Befehlserweiterung für Floppy- und Dateizugriffe · ***Disk-Manager*** Verwaltet bis zu 6500 Programme · ***CP/M <-> CBM*** Einfacher Datenaustausch zwischen CP/M- und 1541-Format · sowie weitere Super-Utilities

Hardware-Voraussetzung: C64/C128, Floppy 1541/1570/1571
Bestell-Nr. 38706 sFr 44,-*/öS 490,-* **DM 49,-***

The Best of Floppy Tools Volume 2

Lieferumfang: Anleitungsheft · 5¼"-Diskette (1541-Format)
ProDisc Komfortable Diskettenverwaltung in Assembler · ***Master-Copy*** Backup-Kopierprogramm · ***Dual-Filecopy*** File-Kopierprogramm für zwei Laufwerke · ***Trackcopy*** Einfaches Kopieren und Formatieren von einzelnen Tracks · ***Tornado-Copy*** Schnelles Backup-Programm für einseitig bespielte Disketten · ***Disk Mon 64*** Professioneller Floppy- und Diskettenmonitor · ***Hypra-Load/Save*** Software-Speeder für C64 · ***Hypra-Perfekt*** Hypra-LOAD/SAVE eingebunden ins Betriebssystem · ***EXOS V3*** Leistungsfähiges Betriebssystem für den C64 · ***Disk-Searcher*** Findet sehr schnell beliebige Zeichenketten auf Diskette · ***File-Manager*** Befehlserweiterung zur Verwaltung von Disketten · ***Super-Autostart*** Autostart-Generator mit Verschlüsselungs-Routine · sowie weitere Super-Utilities

Hardware-Voraussetzung: C64/C128, Floppy 1541/1570/1571
Bestell-Nr. 38707 sFr 44,-*/öS 490,-* **DM 49,-***



Markt&Technik
Zeitschriften · Bücher
Software · Schulung

Dieses Markt&Technik-Softwareprodukt erhalten Sie in den Fachabteilungen der Warenhäuser, in Computer-Fachgeschäften oder im Buchhandel.

710267

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.

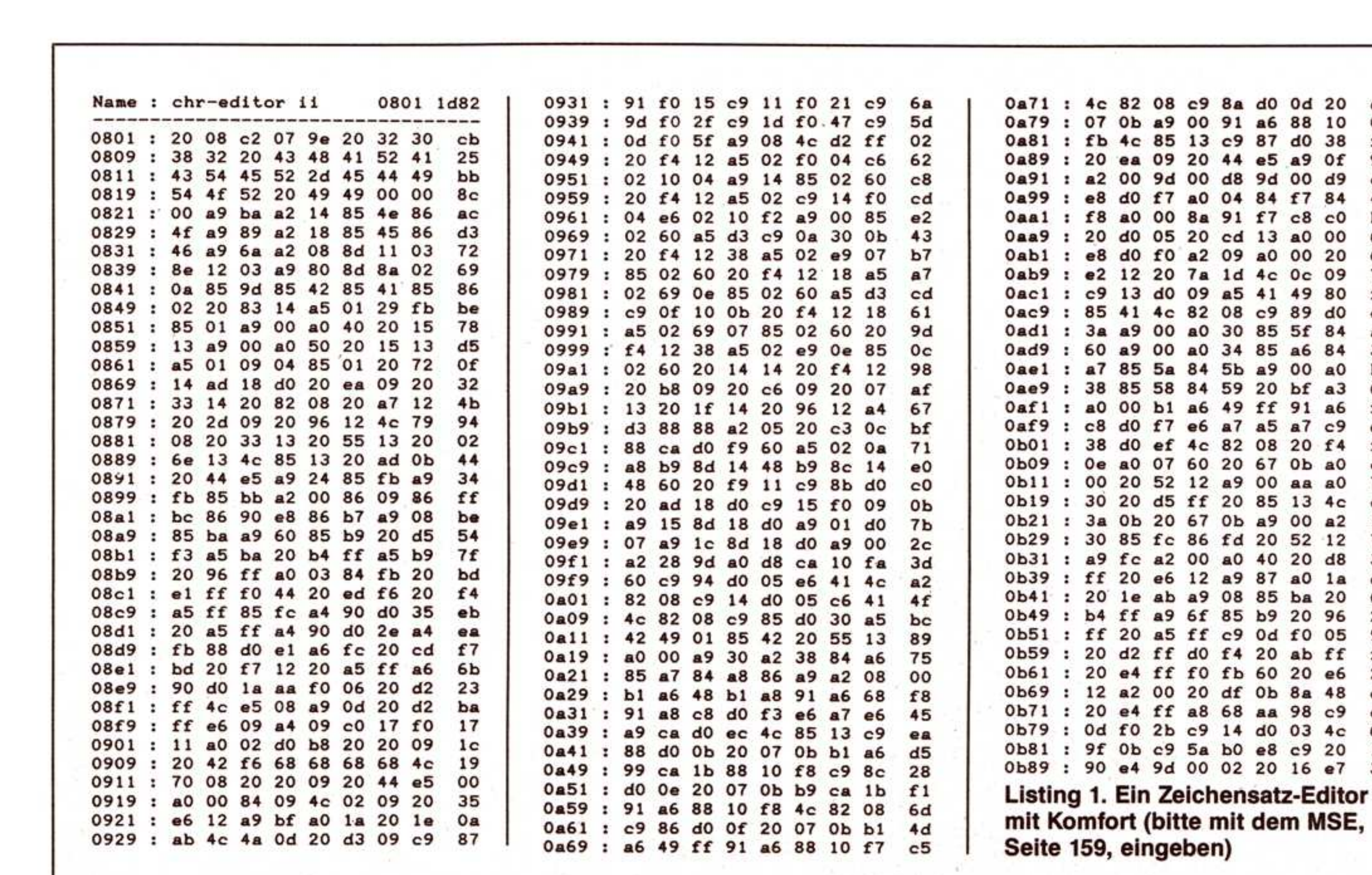
- Wenn Sie den Character-Code 255 um 1 erhöhen, wird er zum Character-Code 0. Ebenso hat ein Erniedrigen des Codes 0 um 1 einen Code von 255 zur Folge.
- Ist der momentan bearbeitete Zeichensatz nicht aktiviert, wird dies durch eine entsprechende Meldung im oberen Teil des Bildschirms angezeigt.
- Sind Eingaben erforderlich, erscheint in der untersten Bildschirmzeile »>« und ein blauer Stern als Cursor. Diese Eingabezeile faßt maximal 35 Zeichen. Steuerzeichen (außer DELETE) und Grafikzeichen werden nicht beachtet. Eine leere Eingabe bewirkt einen Sprung zurück in den Menümodus.
- Fehler bei Zugriffen auf das Diskettenlaufwerk werden abgefangen und angezeigt. Es wird überprüft, ob das Gerät mit der Laufwerksadresse #8 zur Verfügung steht. Wenn nicht, erfolgt eine entsprechende Fehlermeldung. Nach Beendigung einer Diskettenoperation wird der Fehlerkanal ausgelesen und angezeigt.
- Wenn man in Maschinensprache einen dieser neuen Zeichensätze verwenden will, kann man die vom Editor mit der SVC-Routine gespeicherten Zeichensätze laden und mit der Befehlsfolge »LDA #$1C, STA $D018« einschalten und mit der Befehlsfolge »LDA #$15, STA $D018« ausschalten. Es muß nur darauf geachtet werden, daß der Bereich von $3000 bis $3FFF nicht verwendet wird.

Hinweise zum erzeugten Zeichensatz der CCP-Routine

- Sie verfügen über zwei Zeichensätze, die Sie wie gewohnt mit <SHIFT CBM> umschalten können.
- Das Blockieren beziehungsweise Freigeben der Kombination <SHIFT CBM> zum Umschalten der Zeichensätze ist durch die Befehle PRINT CHR$(8) und PRINT CHR$(9) möglich.
- Das Umschalten der Zeichensätze durch die Befehle PRINT CHR$(14) und PRINT CHR$(142) ist nicht möglich!
- Nach einem Reset kann der Zeichensatz mit SYS 51200 erneut aktiviert werden.
- Durch den Befehl SYS 51307 läßt sich der Zeichensatz von Basic aus umschalten.
- Der Bildschirm liegt bei $CC00 bis $CFF7 (52224 bis 53239).
- Die Sprite-Pointer befinden sich bei $CFF8 bis $CFFF (53240 bis 53247).
- Die Aktivierungsroutine liegt bei $C800 bis $C89D (51200 bis 51357).
- Der Bereich von $C000 bis $CFFF (49152 bis 53239) ist nicht mehr verwendbar.
- Der aktive Zeichensatz liegt im Bereich von $C000 bis $C7FF (49152 bis 51199).
- Der inaktive Zeichensatz liegt – vor Basic-Programmen geschützt – am Basic-RAM-Ende. (Dieses wird beim Starten des Zeichensatzprogramms selbständig ermittelt.)
- Der NMI-Vektor $0318/$0319 (792/793) wird verändert und zeigt auf $C800 (51200)
- Wird eine Basic-Erweiterung zusammen mit dem Character-Programm benutzt, muß zuerst diese Erweiterung aktiviert werden und dann das Zeichensatzprogramm! Die benutzte Erweiterung darf den Bereich von $C000 bis $CFFF (49152 bis 53239) nicht verwenden!

Ein Tip zum Abschluß: Sollte irgendein verwendetes Programm (meist Programmierhilfen und Basic-Erweiterungen) den Vektor $028F/$0290 (dezimal 655/656), der den Zeiger auf die Tastaturtabelle setzt, verändern, so ist eine Umschaltung der Zeichensätze durch <SHIFT CBM> nicht möglich. Es erscheint ein undefinierbares Zeichengewirr auf dem Bildschirm, da der VIC durch das »reguläre Umschalten« einen Speicherbereich als Zeichensatz adressiert, der (meist) keine sinnvollen Informationen enthält.

(Reinhard Bartel/sk)



```
Name : chr-editor ii      0801 1d82
-----------------------------------
0801 : 20 08 c2 07 9e 20 32 30  cb
0809 : 38 32 20 43 48 41 52 41  25
0811 : 43 54 45 52 2d 45 44 49  bb
0819 : 54 4f 52 20 49 49 00 00  8c
0821 : 00 a9 ba a2 14 85 4e 86  ac
0829 : 4f a9 89 a2 18 85 45 86  d3
0831 : 46 a9 6a a2 08 8d 11 03  72
0839 : 8e 12 03 a9 80 8d 8a 02  69
0841 : 0a 85 9d 85 42 85 41 85  86
0849 : 02 20 83 14 a5 01 29 fb  be
0851 : 85 01 a9 00 a0 40 20 15  78
0859 : 13 a9 00 a0 50 20 15 13  d5
0861 : a5 01 09 04 85 01 20 72  0f
0869 : 14 ad 18 d0 20 ea 09 20  32
0871 : 33 14 20 82 08 20 a7 12  4b
0879 : 20 2d 09 20 96 12 4c 79  94
0881 : 08 20 33 13 20 55 13 20  02
0889 : 6e 13 4c 85 13 20 ad 0b  44
0891 : 20 44 e5 a9 24 85 fb a9  34
0899 : fb 85 bb a2 00 86 09 86  ff
08a1 : bc 86 90 e8 86 b7 a9 08  be
08a9 : 85 ba a9 60 85 b9 20 d5  54
08b1 : f3 a5 ba 20 b4 ff a5 b9  7f
08b9 : 20 96 ff a0 03 84 fb 20  bd
08c1 : e1 ff f0 44 20 ed f6 20  f4
08c9 : a5 ff 85 fc a4 90 d0 35  eb
08d1 : 20 a5 ff a4 90 d0 2e a4  ea
08d9 : fb 88 d0 e1 a6 fc 20 cd  f7
08e1 : bd 20 f7 12 20 a5 ff a6  6b
08e9 : 90 d0 1a aa f0 06 20 d2  23
08f1 : ff 4c e5 08 a9 0d 20 d2  ba
08f9 : ff e6 09 a4 09 c0 17 f0  17
0901 : 11 a0 02 d0 b8 20 20 09  1c
0909 : 20 42 f6 68 68 68 68 4c  19
0911 : 70 08 20 20 09 20 44 e5  00
0919 : a0 00 84 09 4c 02 09 20  35
0921 : e6 12 a9 bf a0 1a 20 1e  0a
0929 : ab 4c 4a 0d 20 d3 09 c9  87
0931 : 91 f0 15 c9 11 f0 21 c9  6a
0939 : 9d f0 2f c9 1d f0 47 c9  5d
0941 : 0d f0 5f a9 08 4c d2 ff  02
0949 : 20 f4 12 a5 02 f0 04 c6  62
0951 : 02 10 04 a9 14 85 02 60  c8
0959 : 20 f4 12 a5 02 c9 14 f0  cd
0961 : 04 e6 02 10 f2 a9 00 85  e2
0969 : 02 60 a5 d3 c9 0a 30 0b  43
0971 : 20 f4 12 38 a5 02 e9 07  b7
0979 : 85 02 60 20 f4 12 18 a5  a7
0981 : 02 69 0e 85 02 60 a5 d3  cd
0989 : c9 0f 10 0b 20 f4 12 18  61
0991 : a5 02 69 07 85 02 60 20  9d
0999 : f4 12 38 a5 02 e9 0e 85  0c
09a1 : 02 60 20 14 14 20 f4 12  98
09a9 : 20 b8 09 20 c6 09 20 07  af
09b1 : 13 20 1f 14 20 96 12 a4  67
09b9 : d3 88 88 a2 05 20 c3 0c  bf
09c1 : 88 ca d0 f9 60 a5 02 0a  71
09c9 : a8 b9 8d 14 48 b9 8c 14  e0
09d1 : 48 60 20 f9 11 c9 8b d0  c0
09d9 : 20 ad 18 d0 c9 15 f0 09  0b
09e1 : a9 15 8d 18 d0 a9 01 d0  7b
09e9 : 07 a9 1c 8d 18 d0 a9 00  2c
09f1 : a2 28 9d a0 d8 ca 10 fa  3d
09f9 : 60 c9 94 d0 05 e6 41 4c  a2
0a01 : 82 08 c9 14 d0 05 c6 41  4f
0a09 : 4c 82 08 c9 85 d0 30 a5  bc
0a11 : 42 49 01 85 42 20 55 13  89
0a19 : a0 00 a9 30 a2 38 84 a6  75
0a21 : 85 a7 84 a8 86 a9 a2 08  00
0a29 : b1 a6 48 b1 a8 91 a6 68  f8
0a31 : 91 a8 c8 d0 f3 e6 a7 e6  45
0a39 : a9 ca d0 ec 4c 85 13 c9  ea
0a41 : 88 d0 0b 20 07 0b b1 a6  d5
0a49 : 99 ca 1b 88 10 f8 c9 8c  28
0a51 : d0 0e 20 07 0b b9 ca 1b  f1
0a59 : 91 a6 88 10 f8 4c 82 08  6d
0a61 : c9 86 d0 0f 20 07 0b b1  4d
0a69 : a6 49 ff 91 a6 88 10 f7  c5
0a71 : 4c 82 08 c9 8a d0 0d 20  dd
0a79 : 07 0b a9 00 91 a6 88 10  01
0a81 : fb 4c 85 13 c9 87 d0 38  f3
0a89 : 20 ea 09 20 44 e5 a9 0f  9d
0a91 : a2 00 9d 00 d8 9d 00 d9  c9
0a99 : e8 d0 f7 a0 04 84 f7 84  49
0aa1 : f8 a0 00 8a 91 f7 c8 c0  b8
0aa9 : 20 d0 05 20 cd 13 a0 00  6f
0ab1 : e8 d0 f0 a2 09 a0 00 20  68
0ab9 : e2 12 20 7a 1d 4c 0c 09  72
0ac1 : c9 13 d0 09 a5 41 49 80  f4
0ac9 : 85 41 4c 82 08 c9 89 d0  e9
0ad1 : 3a a9 00 a0 30 85 5f 84  aa
0ad9 : 60 a9 00 a0 34 85 a6 84  35
0ae1 : a7 85 5a 84 5b a9 00 a0  b6
0ae9 : 38 85 58 84 59 20 bf a3  67
0af1 : a0 00 b1 a6 49 ff 91 a6  fb
0af9 : c8 d0 f7 e6 a7 a5 a7 c9  de
0b01 : 38 d0 ef 4c 82 08 20 f4  fa
0b09 : 0e a0 07 60 20 67 0b a0  e0
0b11 : 00 20 52 12 a9 00 aa a0  7f
0b19 : 30 20 d5 ff 20 85 13 4c  e2
0b21 : 3a 0b 20 67 0b a9 00 a2  19
0b29 : 30 85 fc 86 fd 20 52 12  7a
0b31 : a9 fc a2 00 a0 40 20 d8  3f
0b39 : ff 20 e6 12 a9 87 a0 1a  d2
0b41 : 20 1e ab a9 08 85 ba 20  68
0b49 : b4 ff a9 6f 85 b9 20 96  29
0b51 : ff 20 a5 ff c9 0d f0 05  9c
0b59 : 20 d2 ff d0 f4 20 ab ff  fb
0b61 : 20 e4 ff f0 fb 60 20 e6  22
0b69 : 12 a2 00 20 df 0b 8a 48  e2
0b71 : 20 e4 ff a8 68 aa 98 c9  ea
0b79 : 0d f0 2b c9 14 d0 03 4c  6f
0b81 : 9f 0b c9 5a b0 e8 c9 20  1d
0b89 : 90 e4 9d 00 02 20 16 e7  3c
```

Listing 1. Ein Zeichensatz-Editor mit Komfort (bitte mit dem MSE, Seite 159, eingeben)

MSE, Seite 159

```
0b91 : e8 e0 23 90 d6 a9 9d 20  36
0b99 : 16 e7 ca 4c 6c 0b a4 d3  38
0ba1 : c0 03 90 ca b0 ef 86 fb  fd
0ba9 : e0 00 f0 58 a9 01 a0 0f  14
0bb1 : a2 08 20 ba ff 20 c0 ff  bb
0bb9 : 90 15 20 e6 12 a9 02 a0  70
0bc1 : 1b 20 1e ab 20 4a 0d a9  c5
0bc9 : 01 20 c3 ff 4c ad 0b a9  7d
0bd1 : 01 20 c3 ff 20 e6 12 a9  a8
0bd9 : db a0 1a 4c 1e ab 8a 48  0e
0be1 : a9 90 a0 1a 20 1e ab 68  b0
0be9 : aa 60 20 de 12 20 fc 12  e1
0bf1 : 20 de 12 a9 71 a0 1a 20  ff
0bf9 : 1e ab 20 e4 ff f0 fb c9  9c
0c01 : 59 d0 01 60 68 68 60 20  9a
0c09 : eb 0b a2 40 a0 00 84 a6  94
0c11 : 86 a7 a2 30 84 a8 86 a9  14
0c19 : a2 08 b1 a6 91 a8 c8 d0  24
0c21 : f9 e6 a7 e6 a9 ca d0 f2  6e
0c29 : 4c 85 13 20 eb 0b a2 48  33
0c31 : 4c 0d 0c a2 0c a0 1d 20  d6
0c39 : e2 12 20 c1 0c 20 45 1d  76
0c41 : 48 a4 d3 b1 d1 30 03 20  f1
0c49 : c1 0c 68 4c 5d 0c 00 a4  33
0c51 : d3 b1 d1 49 04 91 d1 20  ef
0c59 : 98 0c a9 00 c9 11 d0 09  dc
0c61 : a6 d6 e0 13 f0 d7 20 b6  c9
0c69 : 0c c9 91 d0 09 a6 d6 e0  bb
0c71 : 0c f0 ca 20 b6 0c c9 1d  d9
0c79 : d0 09 a6 d3 e0 24 f0 bd  60
0c81 : 20 b6 0c c9 9d d0 09 a6  0a
0c89 : d3 e0 1d f0 b0 20 b6 0c  31
0c91 : c9 0d d0 a9 4c c1 0c 20  8d
0c99 : a8 13 20 ed 13 a2 08 a0  38
0ca1 : 08 b1 f7 49 2e 4a 4a 4a  9c
0ca9 : 6e 00 30 88 d0 f3 20 cd  fd
0cb1 : 13 ca d0 eb 60 48 20 c1  27
0cb9 : 0c 68 20 d2 ff 20 98 0c  d7
0cc1 : a4 d3 b1 d1 49 80 91 d1  78
0cc9 : 60 20 f4 0c a5 f9 c5 42  be
0cd1 : f0 03 20 10 0a a0 07 b1  72
0cd9 : fd 91 aa 88 10 f9 a5 fa  b8
0ce1 : c5 42 f0 03 20 10 0a a0  50
0ce9 : 07 b1 fb 91 ac 88 10 f9  3d
0cf1 : 4c 82 08 a9 03 a2 40 a0  3d
0cf9 : 50 86 fb 84 fd 85 fc 85  27
0d01 : fe 20 e6 12 a9 97 a0 1a  19
0d09 : 20 1e ab 20 52 0d a5 42  d0
0d11 : 85 f9 20 a8 13 84 aa 85  bb
0d19 : ab a0 07 b1 aa 91 fb 88  45
0d21 : 10 f9 20 e6 12 a9 ab a0  71
0d29 : 1a 20 1e ab 20 52 0d a5  64
0d31 : 42 85 fa 20 a8 13 84 ac  87
0d39 : 85 ad a0 07 b1 ac 91 fd  61
0d41 : 88 10 f9 20 e6 12 4c eb  5c
0d49 : 0b 20 d3 09 c9 0d d0 f9  b7
0d51 : 60 20 d3 09 c9 1d d0 05  aa
0d59 : e6 41 20 82 08 c9 11 d0  ed
0d61 : 05 c6 41 20 82 08 c9 0d  c7
0d69 : d0 e7 60 20 f4 0c a0 07  89
0d71 : b1 fb 91 ac 88 10 f9 4c  a3
0d79 : 82 08 20 c8 0d 20 d3 09  54
0d81 : 20 db 0d c9 1d d0 15 a0  f9
0d89 : 24 b1 d1 48 a2 07 88 b1  eb
0d91 : d1 c8 91 d1 88 ca d0 f6  75
0d99 : 68 20 c3 0d c9 9d d0 15  9b
0da1 : a0 1d b1 d1 48 a2 07 c8  be
0da9 : b1 d1 88 91 d1 c8 ca d0  c7
0db1 : f6 68 20 c3 0d a6 02 e0  2c
0db9 : 0c 10 1e c9 0d d0 be 4c  79
0dc1 : cf 0d 91 d1 4c 98 0c a2  b4
0dc9 : 0c a0 25 20 e2 12 20 10  d2
0dd1 : e5 a0 25 b1 d1 49 3f 91  0d
0dd9 : d1 60 c9 11 d0 0e a6 d6  35
0de1 : e0 13 f0 f5 20 cf 0d e6  c8
0de9 : d6 4c cf 0d c9 91 d0 e9  bb
0df1 : a6 d6 e0 0c f0 e3 20 cf  0a
0df9 : 0d c6 d6 4c cf 0d c9 1d  6f
0e01 : d0 0e a6 d3 e0 24 f0 d1  93
0e09 : 20 2a 0e e6 d3 4c 2a 0e  03
0e11 : c9 9d d0 c5 a6 d3 e0 1d  5c
0e19 : f0 bf 20 2a 0e c6 d3 4c  35
0e21 : 2a 0e a2 14 a0 1d 20 e2  b7
0e29 : 12 20 10 e5 a4 d3 b1 d1  5f
0e31 : 49 3e 91 d1 60 20 23 0e  e8
0e39 : 20 d3 09 20 ff 0d c9 91  3c
0e41 : d0 2b a2 0c 86 d6 20 10  91
0e49 : e5 b1 d1 48 a2 07 8a 48  a1
0e51 : 20 a7 0e b1 d1 48 20 ac  38
0e59 : 0e 68 91 d1 e6 d6 68 aa  56
0e61 : ca d0 eb 20 10 e5 68 20  a4
0e69 : c3 0d 20 a7 0e c9 11 d0  c5
0e71 : 27 20 ac 0e b1 d1 48 a2  a5
0e79 : 07 8a 48 20 ac 0e b1 d1  81
0e81 : 48 20 a7 0e 68 91 d1 c6  6d
0e89 : d6 68 aa ca d0 eb 20 10  a4
0e91 : e5 68 20 c3 0d 20 b1 0e  e0
0e99 : a6 02 e0 0c 10 07 c9 0d  74
0ea1 : d0 96 4c 2a 0e 60 e6 d6  42
0ea9 : 4c 10 e5 c6 d6 4c 10 e5  2b
0eb1 : a2 14 86 d6 4c 10 e5 20  f7
0eb9 : d3 09 c9 1d d0 06 20 d4  8e
0ec1 : 0e 20 82 08 c9 11 d0 06  f5
0ec9 : 20 fc 0e 20 82 08 c9 0d  99
0ed1 : d0 e5 60 20 f4 0e a2 07  08
0ed9 : a0 07 b1 a6 4a 91 a6 3e  86
0ee1 : c2 1b 88 10 f5 ca 10 f0  2d
0ee9 : a0 07 b9 c2 1b 91 a6 88  bd
0ef1 : 10 f8 60 20 a8 13 84 a6  1c
0ef9 : 85 a7 60 20 f4 0e a2 07  c6
0f01 : a0 07 b1 a6 0a 91 a6 7e  2b
0f09 : c2 1b 88 10 f5 ca 10 f0  55
0f11 : 4c e9 0e 20 d3 09 c9 1d  c0
0f19 : d0 03 20 2a 0f c9 9d d0  10
0f21 : 03 20 40 0f c9 0d d0 eb  46
0f29 : 60 20 55 10 8a 48 20 aa  b1
0f31 : 13 20 f7 0e 20 d7 0e 68  de
0f39 : aa e8 d0 f0 4c 4f 10 20  69
0f41 : 55 10 8a 48 20 aa 13 20  2e
0f49 : f7 0e 20 ff 0e 68 aa e8  f0
0f51 : d0 f0 4c 4f 10 20 d3 09  fa
0f59 : c9 1d d0 06 20 86 0f 20  58
0f61 : 82 08 c9 9d d0 06 20 98  fc
0f69 : 0f 20 82 08 c9 11 d0 06  9e
0f71 : 20 c1 0f 20 82 08 c9 91  ec
0f79 : d0 06 20 aa 0f 20 82 08  b6
0f81 : c9 0d d0 d1 60 20 f4 0e  36
0f89 : a0 07 b1 a6 4a 90 02 49  b2
0f91 : 80 91 a6 88 10 f4 60 20  ff
0f99 : f4 0e a0 07 b1 a6 0a 90  37
0fa1 : 02 49 01 91 a6 88 10 f4  93
0fa9 : 60 20 f4 0e a0 00 b1 a6  36
0fb1 : aa c8 b1 a6 88 91 a6 c8  42
0fb9 : c0 07 d0 f5 8a 91 a6 60  80
0fc1 : 20 f4 0e a0 07 b1 a6 aa  e1
0fc9 : 88 b1 a6 c8 91 a6 88 d0  ff
0fd1 : f7 8a 91 a6 60 20 d3 09  af
0fd9 : c9 1d d0 03 20 fa 0f c9  6f
0fe1 : 9d d0 03 20 10 10 c9 11  76
0fe9 : d0 03 20 3c 10 c9 91 d0  02
0ff1 : 03 20 26 10 c9 0d d0 dd  94
0ff9 : 60 20 55 10 8a 48 20 aa  81
1001 : 13 20 f7 0e 20 89 0f 68  3f
1009 : aa e8 d0 f0 4c 4f 10 20  39
1011 : 55 10 8a 48 20 aa 13 20  fe
1019 : f7 0e 20 9b 0f 68 aa e8  44
1021 : d0 f0 4c 4f 10 20 55 10  de
1029 : 8a 48 20 aa 13 20 f7 0e  63
1031 : 20 ad 0f 68 aa e8 d0 f0  10
1039 : 4c 4f 10 20 55 10 8a 48  c6
1041 : 20 aa 13 20 f7 0e 20 c4  79
1049 : 0f 68 aa e8 d0 f0 20 82  6e
1051 : 08 4c 72 14 20 83 14 a2  52
1059 : 00 60 20 23 0e 20 d3 09  39
1061 : 20 ff 0d c9 91 d0 32 a2  ab
1069 : 0c 86 d6 20 10 e5 a2 07  3b
1071 : 8a 48 b1 d1 9d c2 1b 20  62
1079 : a7 0e 68 aa ca 10 f1 20  cc
1081 : ac 0e a2 07 8a 48 bd c2  25
1089 : 1b 91 d1 20 ac 0e 68 aa  17
1091 : ca 10 f1 20 98 0c 20 b1  b2
1099 : 0e c9 11 d0 03 4c 68 10  3f
10a1 : c9 0d d0 b9 4c 2a 0e 20  eb
10a9 : c8 0d 20 d3 09 20 db 0d  95
10b1 : c9 1d d0 1d a2 07 a0 1d  00
10b9 : b1 d1 9d c2 1b c8 ca 10  56
10c1 : f7 a2 07 a0 24 bd c2 1b  51
10c9 : 91 d1 88 ca 10 f7 20 98  31
10d1 : 0c c9 9d d0 03 4c b5 10  cd
10d9 : c9 0d d0 ce 4c cf 0d 20  ee
10e1 : d3 09 c9 1d d0 06 20 0a  21
10e9 : 11 20 82 08 c9 9d d0 03  7f
10f1 : 4c e7 10 c9 11 d0 06 20  5e
10f9 : 2b 11 20 82 08 c9 91 d0  bc
1101 : 03 4c f8 10 c9 0d d0 d7  62
1109 : 60 20 f4 0e a0 07 a2 07  53
1111 : 8e c1 1b b1 a6 4a 08 48  ea
1119 : 98 aa 68 28 3e c2 1b ce  29
1121 : c1 1b 10 f1 88 10 e7 4c  f3
1129 : 3b 11 20 f4 0e a2 07 a0  e7
1131 : 00 b1 a6 9d c2 1b c8 ca  25
1139 : 10 f7 a0 07 b9 c2 1b 91  8f
1141 : a6 88 10 f8 60 20 d3 09  b7
1149 : c9 1d d0 03 20 6a 11 c9  63
1151 : 9d d0 03 20 6a 11 c9 11  94
1159 : d0 03 20 80 11 c9 91 d0  0a
1161 : 03 20 80 11 c9 0d d0 dd  bb
1169 : 60 20 55 10 8a 48 20 aa  f1
1171 : 13 20 f7 0e 20 0d 11 68  d3
1179 : aa e8 d0 f0 4c 4f 10 20  a9
1181 : 55 10 8a 48 20 aa 13 20  6e
1189 : f7 0e 20 2e 11 68 aa e8  26
1191 : d0 f0 4c 4f 10 20 67 0b  8c
1199 : a9 00 a0 08 85 5f 84 60  92
11a1 : a9 d2 a0 1b 85 5a 84 5b  33
11a9 : a9 00 a0 70 20 2c 13 a9  8b
11b1 : 2a a0 1c 85 5f 84 60 a9  d2
11b9 : 45 a0 1d 85 5a 84 5b a9  d1
11c1 : 1c a0 09 20 2c 13 a6 fb  62
11c9 : ca e0 0f 30 02 a2 0f bd  ba
11d1 : 00 02 9d 0c 08 ca 10 f7  c2
11d9 : a6 fb e8 e8 a4 fb 86 fb  10
11e1 : b9 00 02 9d 00 02 ca 88  1b
11e9 : 10 f6 a9 2f 9d 00 02 ca  3c
11f1 : a9 43 9d 00 02 4c d2 1b  a7
11f9 : 20 e4 ff a2 13 dd 30 12  e4
1201 : f0 2c ca 10 f8 20 70 1d  48
1209 : a2 07 a0 25 48 20 e2 12  31
1211 : 68 20 d2 ff ad 3d 05 85  22
1219 : 41 a9 07 8d 3d d9 a9 00  ec
1221 : 85 d4 ae 82 1d ac 83 1d  8c
1229 : 20 0c e5 4c 82 08 60 00  3c
1231 : 0d 11 12 13 14 1d 91 92  43
1239 : 93 94 9d 85 86 87 88 89  08
1241 : 8a 8b 8c 00 00 00 00 ea  8a
1249 : ea ea ea ea ea ea ea ea  48
1251 : 00 a2 08 20 ba ff a5 fb  e3
1259 : a2 00 a0 02 4c bd ff 20  57
1261 : 67 0b a9 08 a0 6f 20 52  64
1269 : 12 20 c0 ff a9 08 20 c3  9e
1271 : ff 4c 3a 0b 20 eb 0b 68  e5
1279 : 68 68 68 20 44 e5 a9 15  78
1281 : 8d 18 d0 20 74 a4 60 a5  8c
1289 : 02 0a a8 b1 4e aa c8 b1  b1
1291 : 4e a8 4c e2 12 20 88 12  0b
1299 : a9 0d 8d 86 02 20 ef 12  02
12a1 : a5 02 c5 a4 f0 e0 85 a4  c3
12a9 : 20 14 14 a9 03 8d 86 02  c8
12b1 : a9 12 20 d2 ff a2 17 a0  78
12b9 : 00 20 e2 12 20 00 13 a2  58
12c1 : 17 a0 02 20 e2 12 a5 02  06
12c9 : 0a a8 b1 45 48 c8 b1 45  58
12d1 : a8 68 20 1e ab a9 92 20  0c
12d9 : d2 ff 4c 1f 14 a2 18 a0  9a
12e1 : 02 18 4c f0 ff 20 de 12  c1
12e9 : 20 fc 12 4c de 12 a9 5f  79
12f1 : 4c d2 ff 20 88 12 a9 20  aa
12f9 : 4c d2 ff a9 00 85 d3 a9  b2
1301 : 48 a0 1a 4c 1e ab a2 18  a3
1309 : a0 00 20 e2 12 a9 61 a0  43
1311 : 18 4c 1e ab 85 58 84 59  2c
1319 : a9 00 a0 d0 85 5f 84 60  2b
1321 : a9 00 a0 e0 85 5a 84 5b  02
1329 : 4c bf a3 85 58 84 59 4c  96
1331 : bf a3 20 14 14 a9 07 8d  12
1339 : 86 02 20 4e 13 20 04 14  fd
1341 : 20 4e 13 a9 00 a6 41 20  fd
1349 : cd bd 4c 1f 14 a2 07 a0  a0
1351 : 09 4c e2 12 20 14 14 a9  c2
1359 : 07 8d 86 02 a2 07 a0 17  1c
1361 : 20 e2 12 a9 00 a6 42 20  2b
1369 : cd bd 4c 1f 14 20 14 14  c7
1371 : a9 07 8d 86 02 a2 07 a0  65
1379 : 25 20 e2 12 a5 41 8d 3d  be
1381 : 05 4c 1f 14 20 a8 13 20  cb
1389 : ed 13 a2 08 ad 00 30 a0  86
1391 : 08 4a b0 06 20 f6 13 4c  4a
1399 : 9e 13 20 fd 13 88 d0 f1  25
13a1 : 20 cd 13 ca d0 e6 60 a5  d7
13a9 : 41 a2 00 85 71 86 72 a2  46
13b1 : 03 06 71 26 72 ca d0 f9  8d
13b9 : a4 71 8c 8e 13 8c aa 0c  63
13c1 : 18 a9 30 65 72 8d 8f 13  5f
13c9 : 8d ab 0c 60 18 a5 f7 69  9c
13d1 : 28 85 f7 90 02 e6 f8 18  37
13d9 : ad 8e 13 69 01 8d 8e 13  9c
13e1 : 8d aa 0c 90 06 ee 8f 13  15
13e9 : ee ab 0c 60 a2 fc a0 05  5b
13f1 : 86 f7 84 f8 60 48 a9 2e  fe
13f9 : 91 f7 68 60 48 a9 2a 91  4a
1401 : f7 68 60 a5 d3 85 b4 a9  89
1409 : b6 a0 14 20 1e ab a6 b4  5c
1411 : 86 d3 60 38 20 f0 ff 86  37
1419 : b2 84 b3 4c 2d 14 a6 b2  f7
1421 : a4 b3 18 20 f0 ff a5 07  5d
1429 : 8d 86 02 60 ad 86 02 85  a8
1431 : 07 60 20 7b 14 a9 00 8d  8a
1439 : 20 d0 8d 21 d0 85 0c a9  06
1441 : e4 85 3f a9 14 85 40 e6  29
```

```
1449 : 0c a4 0c 88 b1 3f 48 c8  83
1451 : b1 3f a8 68 20 1e ab e6  48
1459 : 0c a5 0c c9 24 30 e8 ad  37
1461 : 18 d0 c9 15 d0 03 20 e1  60
1469 : 09 ad 11 d0 49 10 8d 11  15
1471 : d0 ad 0e dc 49 01 8d 0e  26
1479 : dc 60 ad 11 d0 29 ef 8d  44
1481 : 11 d0 ad 0e dc 29 fe 8d  56
1489 : 0e dc 60 33 0c c9 0c 6b  9a
1491 : 0d 0c 0b 22 0b 07 0c 2b  1b
1499 : 0c b7 0e 13 0f 35 0e 7a  2f
14a1 : 0d 55 0f d5 0f 5a 10 a7  2b
14a9 : 10 df 10 45 11 8d 08 5f  b2
14b1 : 12 95 11 74 12 20 20 20  44
14b9 : 00 0c 06 0d 06 0e 06 0f  ea
14c1 : 06 10 06 11 06 12 06 0c  94
14c9 : 0c 0d 0c 0e 0c 0f 0c 10  aa
14d1 : 0c 11 0c 12 0c 0c 12 0d  2f
14d9 : 12 0e 12 0f 12 10 12 11  65
14e1 : 12 12 12 08 15 32 15 5c  72
14e9 : 15 86 15 ae 15 d2 15 02  9d
14f1 : 16 2f 16 5f 16 94 16 c9  02
14f9 : 16 fe 16 33 17 68 17 9d  c7
1501 : 17 d2 17 05 18 34 18 93  93
1509 : 20 20 20 20 12 98 b0 c0  70
1511 : c0 c0 c0 c0 c0 c0 c0 c0  10
1519 : c0 c0 c0 c0 c0 c0 c0 c0  18
1521 : c0 c0 c0 c0 c0 c0 c0 c0  20
1529 : c0 c0 c0 c0 c0 ae 92 0d  77
1531 : 00 20 20 20 20 12 dd 1e  94
1539 : 20 3c 49 49 3e 20 20 43  df
1541 : 48 41 52 41 43 54 45 52  77
1549 : 2d 45 44 49 54 4f 52 20  9c
1551 : 20 3c 49 49 3e 20 98 dd  0e
1559 : 92 0d 00 20 20 20 20 12  1e
1561 : dd 96 20 20 28 43 29 31  39
1569 : 39 38 36 20 42 59 20 52  64
1571 : 45 49 4e 48 41 52 44 20  ef
1579 : 42 41 52 54 45 4c 20 20  f2
1581 : 98 dd 92 0d 00 20 20 20  10
1589 : 20 12 ad c0 c0 c0 c0 c0  cc
1591 : c0 c0 c0 c0 c0 c0 c0 c0  90
1599 : c0 c0 c0 c0 c0 c0 c0 c0  98
15a1 : c0 c0 c0 c0 c0 c0 c0 c0  a0
15a9 : c0 bd 92 0d 00 90 20 20  d3
15b1 : 20 20 20 20 5a 45 49 43  69
15b9 : 48 45 4e 53 41 54 5a 20  02
15c1 : 4e 49 43 48 54 20 41 4b  70
15c9 : 54 49 56 49 45 52 54 0d  d3
15d1 : 00 98 11 11 20 43 48 52  66
15d9 : 2d 4e 52 3a 20 20 20 20  cd
15e1 : 20 20 43 48 52 2d 53 45  52
15e9 : 54 3a 20 20 20 20 43 48  07
15f1 : 41 52 41 43 54 45 52 3a  41
15f9 : 20 9e 20 20 98 11 11 0d  e5
1601 : 00 20 20 20 20 20 20 20  e1
1609 : 20 20 20 20 20 20 20 20  09
1611 : 20 20 20 20 20 20 20 20  11
1619 : 20 20 20 20 9e 12 b0 c0  14
1621 : c0 c0 c0 c0 c0 c0 c0 c0  20
1629 : c0 ae 92 9f 0d 00 9f b0  8a
1631 : c0 c0 c0 c0 c0 c0 c0 c0  30
1639 : c0 c0 c0 c0 c0 c0 c0 c0  38
1641 : c0 c0 ae 20 20 20 20 20  d5
1649 : 20 20 9e 12 dd 9a 20 31  f9
1651 : 32 33 34 35 36 37 38 20  0f
1659 : 9e dd 92 9f 0d 00 dd 96  f4
1661 : 20 45 44 43 20 20 20 54  ca
1669 : 43 48 20 20 20 4d 42 56  ff
1671 : 20 20 9f dd 20 20 20 20  09
1679 : 20 20 20 9e 12 dd 9a 31  62
1681 : 92 9e 2e 2e 2e 2e 2e 2e  1d
1689 : 2e 2e 12 9a 20 9e dd 92  3a
1691 : 9f 0d 00 dd 96 20 43 48  7b
1699 : 43 20 20 20 54 43 53 20  e5
16a1 : 20 20 4d 43 48 20 20 9f  d2
16a9 : dd 20 20 20 20 20 20 20  66
16b1 : 9e 12 dd 9a 32 92 9e 2e  b2
16b9 : 2e 2e 2e 2e 2e 2e 2e 12  81
16c1 : 9a 20 9e dd 92 9f 0d 00  29
16c9 : dd 96 20 4d 56 43 20 20  e3
16d1 : 20 52 42 48 20 20 20 4d  d2
16d9 : 43 53 20 20 9f dd 20 20  7c
16e1 : 20 20 20 20 20 9e 12 dd  18
16e9 : 9a 33 92 9e 2e 2e 2e 2e  ff
16f1 : 2e 2e 2e 2e 12 9a 20 9e  3b
16f9 : dd 92 9f 0d 00 dd 96 20  32
1701 : 4c 44 43 20 20 20 52 42  15
1709 : 56 20 20 20 44 49 52 20  94
1711 : 20 9f dd 20 20 20 20 20  40
1719 : 20 20 9e 12 dd 9a 34 92  dc
1721 : 9e 2e 2e 2e 2e 2e 2e 2e  91
1729 : 2e 12 9a 20 9e dd 92 9f  6d
1731 : 0d 00 dd 96 20 53 56 43  05
1739 : 20 20 20 52 43 48 20 20  f3
1741 : 20 53 44 43 20 20 9f dd  c2
1749 : 20 20 20 20 20 20 20 9e  46
1751 : 12 dd 9a 35 92 9e 2e 2e  d2
1759 : 2e 2e 2e 2e 2e 2e 12 9a  c2
1761 : 20 9e dd 92 9f 0d 00 dd  b8
1769 : 96 20 4c 53 31 20 20 20  62
1771 : 52 43 53 20 20 20 43 43  d4
1779 : 50 20 20 9f dd 20 20 20  75
1781 : 20 20 20 20 9e 12 dd 9a  e5
1789 : 36 92 9e 2e 2e 2e 2e 2e  df
1791 : 2e 2e 2e 12 9a 20 9e dd  85
1799 : 92 9f 0d 00 dd 96 20 4c  ea
17a1 : 53 32 20 20 20 4d 42 48  20
17a9 : 20 20 20 4c 43 45 20 20  8a
17b1 : 9f dd 20 20 20 20 20 20  0f
17b9 : 20 9e 12 dd 9a 37 92 9e  53
17c1 : 2e 2e 2e 2e 2e 2e 2e 2e  c1
17c9 : 12 9a 20 9e dd 92 9f 0d  0f
17d1 : 00 ad c0 c0 c0 c0 c0 c0  87
17d9 : c0 c0 c0 c0 c0 c0 c0 c0  d8
17e1 : c0 c0 c0 c0 bd 20 20 20  e7
17e9 : 20 20 20 20 9e 12 dd 9a  4d
17f1 : 38 92 9e 2e 2e 2e 2e 2e  49
17f9 : 2e 2e 2e 12 9a 20 9e dd  ed
1801 : 92 9f 0d 00 20 20 20 20  6a
1809 : 20 20 20 20 20 20 20 20  09
1811 : 20 20 20 20 20 20 20 20  11
1819 : 20 20 20 20 20 20 20 9e  16
1821 : 12 dd 9a 20 20 20 20 20  90
1829 : 20 20 20 20 20 9e dd 92  f9
1831 : 9f 0d 00 20 20 20 20 20  1f
1839 : 20 20 20 20 20 20 20 20  39
1841 : 20 20 20 20 20 20 20 20  41
1849 : 20 20 20 20 20 20 9e 12  27
1851 : ad c0 c0 c0 c0 c0 c0 c0  3d
1859 : c0 c0 c0 bd 92 98 0d 00  83
1861 : 20 20 20 20 20 20 20 20  61
1869 : 20 20 20 20 20 20 20 20  69
1871 : 20 20 20 20 20 20 20 20  71
1879 : 20 20 20 20 20 20 20 20  79
1881 : 20 20 20 20 20 20 20 00  41
1889 : b3 18 c2 18 d4 18 e3 18  ca
1891 : f6 18 09 19 1d 19 31 19  8a
1899 : 40 19 53 19 6a 19 7f 19  fd
18a1 : 90 19 a5 19 bc 19 d1 19  58
18a9 : e2 19 f7 19 06 1a 18 1a  ff
18b1 : 31 1a 45 44 49 54 20 43  07
18b9 : 48 41 52 41 43 54 45 52  ef
18c1 : 00 43 48 41 4e 47 45 20  12
18c9 : 43 48 41 52 41 43 54 45  d5
18d1 : 52 53 00 4d 4f 56 45 20  74
18d9 : 43 48 41 52 41 43 54 45  e5
18e1 : 52 00 4c 4f 41 44 20 43  6e
18e9 : 48 41 52 41 43 54 45 52  1f
18f1 : 2d 53 45 54 00 53 41 56  f0
18f9 : 45 20 43 48 41 52 41 43  5a
1901 : 54 45 52 2d 53 45 54 00  e3
1909 : 4c 4f 41 44 20 43 48 41  95
1911 : 52 41 43 54 45 52 2d 53  a1
1919 : 45 54 31 00 4c 4f 41 44  a2
1921 : 20 43 48 41 52 41 43 54  02
1929 : 45 52 2d 53 45 54 32 00  0d
1931 : 54 55 52 4e 20 43 48 41  4e
1939 : 52 41 43 54 45 52 00 54  17
1941 : 55 52 4e 20 43 48 41 52  77
1949 : 41 43 54 45 52 2d 53 45  50
1951 : 54 00 52 4f 54 41 54 45  4f
1959 : 20 42 59 54 45 20 48 4f  90
1961 : 52 49 5a 4f 4e 54 41 4c  fd
1969 : 00 52 4f 54 41 54 45 20  fd
1971 : 42 59 54 45 20 56 45 52  8c
1979 : 54 49 43 41 4c 00 52 4f  18
1981 : 54 41 54 45 20 43 48 41  f3
1989 : 52 41 43 54 45 52 00 52  63
1991 : 4f 54 41 54 45 20 43 48  d8
1999 : 41 52 41 43 54 45 52 2d  cf
19a1 : 53 45 54 00 4d 49 52 52  b9
19a9 : 4f 52 20 42 59 54 45 20  ff
19b1 : 48 4f 52 49 5a 4f 4e 54  61
19b9 : 41 4c 00 4d 49 52 52 4f  d9
19c1 : 52 20 42 59 54 45 20 56  7c
19c9 : 45 52 54 49 43 41 4c 00  e5
19d1 : 4d 49 52 52 4f 52 20 43  30
19d9 : 48 41 52 41 43 54 45 52  0f
19e1 : 00 4d 49 52 52 4f 52 20  4e
19e9 : 43 48 41 52 41 43 54 45  f5
19f1 : 52 2d 53 45 54 00 44 49  40
19f9 : 53 4b 2d 44 49 52 45 43  88
```

Listing 1. (Fortsetzung)







Comic: Ingo Stein

```
1a01 : 54 4f 52 59 00 53 45 4e   09
1a09 : 44 20 44 49 53 4b 2d 43   62
1a11 : 4f 4d 4d 41 4e 44 00 43   10
1a19 : 52 45 41 54 45 20 43 48   dc
1a21 : 41 52 41 43 54 45 52 2d   57
1a29 : 50 52 4f 47 52 41 4d 00   c3
1a31 : 4c 45 41 56 45 20 43 48   2e
1a39 : 41 52 41 43 54 45 52 2d   6f
1a41 : 45 44 49 54 4f 52 00 99   40
1a49 : 3d 3e 20 20 20 20 20 20   75
1a51 : 20 20 20 20 20 20 20 20   51
1a59 : 20 20 20 20 20 20 20 20   59
1a61 : 20 20 20 20 20 20 20 20   61
1a69 : 20 20 20 20 20 20 20 00   29
1a71 : 41 52 45 20 59 4f 55 20   d6
1a79 : 53 48 55 52 45 20 28 59   39
1a81 : 2f 4e 29 3f 20 00 53 54   02
1a89 : 41 54 55 53 3a 20 00 9a   8e
1a91 : 2a 20 9d 9d 05 00 53 45   0f
1a99 : 4c 45 43 54 20 31 2e 20   68
1aa1 : 43 48 41 52 41 43 54 45   ad
1aa9 : 52 00 53 45 4c 45 43 54   1d
1ab1 : 20 32 2e 20 43 48 41 52   9a
1ab9 : 41 43 54 45 52 00 50 52   65
1ac1 : 45 53 53 20 27 52 45 54   4b
1ac9 : 55 52 4e 27 20 54 4f 20   e2
1ad1 : 43 4f 4e 54 49 4e 55 45   c1
1ad9 : 9b 00 9b 44 49 53 4b 4f   df
1ae1 : 50 45 52 41 54 49 4f 4e   fa
1ae9 : 20 49 4e 20 50 52 4f 43   a1
1af1 : 45 53 53 2e 20 50 4c 45   bb
1af9 : 41 53 45 20 57 41 49 54   87
1b01 : 00 1e 12 50 4c 45 41 53   3a
1b09 : 45 20 53 57 49 54 43 48   f3
1b11 : 20 44 52 49 56 45 20 23   67
1b19 : 38 20 4f 4e 20 21 21 21   d1
1b21 : 20 20 20 20 20 20 20 20   21
1b29 : 92 00 00 1d 08 c1 07 9e   47
1b31 : 20 32 30 37 39 20 20 20   b3
1b39 : 20 20 20 20 20 20 20 20   39
1b41 : 20 20 20 20 20 20 00 00   81
1b49 : 00 a9 40 a0 08 85 5f 84   75
1b51 : 60 a9 96 a0 10 85 5a 84   df
1b59 : 5b a9 56 a0 c8 85 58 84   56
1b61 : 59 20 bf a3 4c 00 c8 00   17
1b69 : 00 00 ad 00 dd 29 fc 8d   0b
1b71 : 00 dd a9 30 8d 18 d0 a9   00
1b79 : cc 8d 88 02 a9 00 a2 0e   af
1b81 : a0 0f 8d 86 02 8e 20 d0   94
1b89 : 8c 21 d0 a9 3c a0 03 8d   ff
1b91 : 81 02 8c 82 02 a9 80 8d   11
1b99 : 91 02 a0 57 a9 c8 84 b2   97
1ba1 : 85 b3 cd 19 03 f0 16 8d   c2
1ba9 : 19 03 a9 00 8d 18 03 20   94
1bb1 : 02 fd d0 06 20 44 e5 4c   fb
1bb9 : 74 a4 6c 00 a0 6c 02 a0   51
1bc1 : 00 00 00 00 00 00 00 00   c2
1bc9 : 00 00 00 00 00 00 00 00   ca
1bd1 : 00 a9 00 a0 30 85 5f 84   70
1bd9 : 60 a9 00 a0 40 85 5a 84   c5
1be1 : 5b a9 1b a0 19 85 58 84   14
1be9 : 59 20 bf a3 a9 01 a2 08   f4
1bf1 : 85 fc 86 fd a2 08 20 ba   b6
1bf9 : ff a5 fb a2 00 a0 02 20   6b
1c01 : bd ff a9 fc a2 20 a0 19   a8
1c09 : 20 d8 ff a9 2e a0 5c 85   2f
1c11 : 5f 84 60 a9 00 a0 70 85   d1
1c19 : 5a 84 5b a9 d2 a0 1b 85   6b
1c21 : 58 84 59 20 bf a3 4c 3a   d4
1c29 : 0b 1c 08 c2 07 9e 20 32   e7
1c31 : 30 37 39 20 20 20 20 20   13
1c39 : 20 20 20 20 20 20 20 20   39
1c41 : 20 20 20 20 00 00 00 a9   d1
1c49 : 1b a2 11 85 5f 86 60 a9   a9
1c51 : 1b a2 19 85 5a 86 5b c6   89
1c59 : 38 a9 00 a6 38 85 58 86   59
1c61 : 59 20 bf a3 a9 1b a2 09   3f
1c69 : 85 5f 86 60 a9 1b a2 11   6c
1c71 : 85 5a 86 5b a9 00 a2 c8   e7
1c79 : 85 58 86 59 20 bf a3 a9   d9
1c81 : 7d a2 08 85 5f 86 60 a9   01
1c89 : 1b a2 09 85 5a 86 5b a9   83
1c91 : 9e a2 c8 85 58 86 59 20   c2
1c99 : bf a3 a5 38 38 e9 08 8d   a8
1ca1 : 9d c8 4c 00 c8 a9 52 a2   1e
1ca9 : c8 8d 8f 02 8e 90 02 ad   2d
1cb1 : 00 dd 29 fc 8d 00 dd a9   2d
1cb9 : 30 8d 18 d0 a9 cc 8d 88   18
1cc1 : 02 a9 00 a2 0e a0 0f 8d   29
1cc9 : 86 02 8e 20 d0 8c 21 d0   90
1cd1 : 85 cf a9 c8 cd 19 03 f0   55
1cd9 : 18 8d 19 03 a9 00 8d 18   60
1ce1 : 03 ad 9c c8 ae 9d c8 85   01
1ce9 : 37 86 38 20 22 e4 4c 74   d9
1cf1 : a4 20 44 e5 6c 02 a0 a9   20
1cf9 : 30 8d 18 d0 ad 8d 02 c9   f3
1d01 : 03 d0 3b cd 8e 02 d0 03   37
1d09 : 4c 42 eb ad 91 02 30 2b   67
1d11 : ad 9c c8 ae 9d c8 85 22   8f
1d19 : 86 23 a9 00 a2 c0 85 24   2a
1d21 : 86 25 a0 00 b1 22 48 b1   13
1d29 : 24 91 22 68 91 24 c8 d0   aa
1d31 : f3 e6 23 e6 25 a5 25 c9   e5
1d39 : c8 d0 e9 4c 76 eb 4c 64   2e
1d41 : eb 00 00 00 a5 d6 85 94   7d
1d49 : a5 d3 85 95 20 d3 09 48   41
1d51 : a6 94 a4 95 20 0c e5 68   e8
1d59 : ae 8d 02 e4 96 f0 0f e0   da
1d61 : 02 d0 09 86 96 68 68 a9   80
1d69 : 00 4c 50 0c 86 96 60 20   04
1d71 : 13 e5 8e 82 1d 8c 83 1d   e9
1d79 : 60 20 cf ff a9 00 85 d4   37
1d81 : 60 00 00 00 00 00 00 00   e2
```

Listing 1. (Schluß)



64'er PROGRAMMSERVICE

Weitere Angebote zum Thema Grafik

Weitere Programmservice-Angebote zu 64'er-Sonderheften

Grafik- und Anwendungsprogramme der Spitzenklasse (SH 23)

Paint Magic: Dieses professionelle Multicolor Mal- und Zeichenprogramm entlockt Ihrem C64 die farbenprächtigsten Bilder. Trotzdem ist es so einfach zu bedienen, daß Sie von der ersten Minute an Ihre Freude an Paint Magic haben werden. **Movie-Show:** Ein Programm, das hochauflösende Grafik zum Leben erweckt. Dank trickreicher Programmierung werden bis zu 99 Hires-Grafiken im Speicher gehalten, so daß Sie beispielsweise perfekte Simulationen von Wellenbewegungen und anderen physikalische Phänomenen erzeugen können. **Boerse Plus:** Schnell reich werden, ohne zu arbeiten – der C64 hilft Ihnen dabei. Börse Plus, programmiert von einem Börsenexperten, ist ein vielseitiges Aktienverwaltungsprogramm. Grafische Anzeige der Kursverläufe, Gewinnberechnungen, professionelle Charts und vieles mehr bietet Börse Plus. Als kostenlose Zugabe befinden sich die Kurse wichtiger AGs der letzten Monate ebenfalls auf der Programmservice-Diskette. Natürlich finden Sie auch in diesem Sonderheft viele Tips & Tricks. Sie finden beispielsweise **»Hardcopy-Programmierung für jeden Drucker«**, ein Programm zum Suchen und Manipulieren von Sprites auf Diskette einen Kurs zur Scroll-Programmierung und vieles mehr. Natürlich enthält die Programmservice-Diskette auch alle Programme, die mit einem Diskettensymbol gekennzeichnet sind.
1 Diskette für C64/C128

Bestell-Nr.: 15723 **DM 29,90*** sFr 24,90*/öS 299,–*

Assembler-Programmierung einfach wie in Basic (SH 21)

Giga-Ass: Ein neu gestalteter Makro-Assembler der Spitzenklasse erlaubt es, Maschinensprache-Programme so komfortabel wie in Basic zu schreiben. Durch ein mitgeliefertes Konvertierungsprogramm können auch Hypra-Ass-Quelltexte mit Giga-Ass verarbeitet werden. Eine große Hilfe beim Schreiben eigener Maschinenprogramme ist auch die Makro-Bibliothek auf Diskette, die oft verwendete Routinen bereitstellt. Auch für den Basic-Programmierer wird einiges geboten: **DMA-Basic:** Mit diesem Programm können Sie die 512-KByte-RAM-Erweiterung von Commodore am C64 verwenden und somit über 576-KByte-Speicherplatz verfügen. **Paradoxon-Basic:** Eine weitere Basic-Erweiterung, stellt dem Basic Programmierer 50 KByte für seine Programme und eine Reihe neuer mächtiger Befehle zur Verfügung. **Rekursiv-Basic:** Diese Erweiterung ist von einer anderen Qualität: Sie erlaubt, wie der Name schon sagt, rekursive Programmierung durch einen vergrößerten Stack und vor allem die Definition von Prozeduren wie in Pascal. **Obsess V3.1:** Dieses Programm erlaubt es, auf einfache Weise mit normalen Matrixdruckern Multicolor-Grafiken in strahlenden Farben auch zu Papier zu bringen. Ergänzt werden die Programme durch eine Reihe von Tips&Tricks. Für den ambitionierten Programmierer befinden sich des weiteren sämtliche Quell-Codes der Maschinenprogramme auf Diskette.
1 Diskette für C64/ C128

Bestell-Nr.: 15721 **DM 29,90*** sFr 24,90*/öS 299,–*

Mastertext 128 – kaum zu übertreffen (SH 18)

Master-Text 128: Das super-professionelle Master-Text für den C128 bietet professionelle Leistungsmerkmale. Durch Menü- und Window-Steuerung ist das Programm anwenderfreundlich und bietet neben dem Standard an Befehlen noch Textbaustein-Funktionen, einen Terminal-Modus, einen integrierten Taschenrechner sowie eine Uhr mit Alarmfunktion. **Textas:** Das Textverarbeitungsprogramm Textas läuft auf den C64 und ist speziell für den MPS 801 und kompatible Drucker entwickelt worden. Mit selbstdefinierbarem Zeichensatz stellt Textas nun auch die deutschen Sonderzeichen zur Verfügung, bietet eine deutlich bessere Druckqualität und erlaubt sogar das Einbinden von Hires-Grafiken und Sprites. **MacMatrix:** Mit MacMatrix stellen wir Ihnen ein Programm zur Verfügung, mit dem Sie auf komfortable Art und Weise NLQ-Zeichensätze für Ihren NL-10 mit Commodore-Interface entwerfen können. Außerdem sind noch viele Tips und Tricks, zum Beispiel für Vizawrite, auf der Diskette enthalten. Die Beschreibungen finden Sie im Sonderheft Ausgabe 18/87 (Drucker)
1 Diskette für C64/C128

Bestell-Nr.: 15718 **DM 29,90*** sFr 24,90*/öS 299,–*

Weitere Programmservice-Angebote zu 64'er-Stammheften:

Korrespondenz-Qualität und Grafik mit Vizawrite 64

Viza-Print-System: Dieses Programm entlockt allen Epson-kompatiblen Druckern eine unübertreffliche Druckqualität. Zehn fertige, professionelle Zeichensätze mit einer 24x25-Punktmatrix können für den Ausdruck von Vizawrite-Texten in Proportionalschrift bei gleichzeitigem Blocksatz und Grafikverbund eingesetzt werden. Die druckfertigen Demo-Texte auf der Diskette helfen Ihnen, mit dem V-P-System schnell vertraut zu werden. **Mony 64:** Mony 64 macht die Verwaltung Ihrer Einnahmen und Ausgaben zum Kinderspiel, sogar Fixbuchungen werden automatisch durchgeführt. **Super-Hardcopy:** Super-Hardcopy kann jeden Bildschirm, der gerade angezeigt wird, auf Tastendruck ausdrucken, auch während eines laufenden Programms. **Hardcopy für Seikosha SP-1000:** Dieses Programm ermöglicht es Hardcopies auf dem Seikosha SP-1000 mit 960 Punkten pro Zeile auf das Papier zu bringen. **PFox+:** Durch PFox+ wird die Qualität Ihrer Printfox-Ausdrucke auf allen Epson-kompatiblen Druckern erhöht. **Unterlängen:** Das Programm Unterlängen ermöglicht es Ihnen, mit Ihrem MPS 801-Drucker problemlos deutsche Umlaute, Superscript, Unterlängen, deutsche Anführungszeichen und Unterstreichungen darzustellen. **Stereochanger:** Dieses Programm schreibt selbständig fast jedes Musikstück, das im Interrupt des C64 läuft, für unseren »Stereo-SID« aus der Ausgabe 11/86 um. Der Raumklang ist fantastisch! Außerdem noch viele Tips und Tricks für den C64/C128/C16 und Plus/4. Die Beschreibung finden Sie in Ausgabe 6/87
Diskette für C64

Bestell-Nr.: 10706 **DM 29,90*** sFr 24,90*/öS 299,–*

Der gute Geist für Ihre Floppy 1541

Disk-Demon: Disk-Demon ist ein Diskettenmonitor ganz besonderer Art. Wenn Sie Probleme mit fehlerhaften Disketten haben, auf denen wichtige Daten gespeichert sind, oder Sie eine Diskette lediglich einmal genauer unter die Lupe nehmen wollen, dann ist der Disk-Demon genau das richtige Werkzeug für Sie. Er liest, analysiert und repariert defekte Sektoren, bearbeitet die Spuren 0 bis 42 auf einer Diskette und unterstützt Sie auch dann, wenn es um die Entwicklung eigener Kopierschutzmethoden geht. **Hi-Eddi+ mit Maus:** Mit diesen Routinen kann man die Proportional-Maus von Reisware für die Steuerung von Hi-Eddi+ verwenden. **Hirn 64:** Hirn 64 ist ein mit Hypra-Basic geschriebenes Spiel, das auch Sie in seinen Bann ziehen wird. Es ist eine grafisch ausgefeilte Variante des bekannten Masterminds, auch als Superhirn bekannt. Die Beschreibung finden Sie in Ausgabe 8/87
Diskette für C64/C128

Bestell-Nr.: 10708 **DM 29,90*** sFr 24,90*/öS 299,–*

Schnell wie der Wind

Master-Copy: Ohne jede Erweiterung in Computer oder Floppystation kopiert Ihnen Master-Copy eine komplette Diskette in weniger als 90 Sekunden; auf dem C64 mit einem Einzellaufwerk ein absoluter Rekord. **Master-Index:** Wenn Sie irgend einen Artikel oder ein Programm aus dem 64'er-Magazin suchen, kann Ihnen dieses Programm anhand eines Stichwortes Ausgabe und Seite des Artikels ermitteln. **Transformer3:** Transformer3 überträgt Texte von Vizawrite 64 auf das Programm Printfox. **Pic-Loader:** Pic-Loader konvertiert die Grafikminiaturen aus Printshop und Printmaster in ein Format, das von Hi-Eddi oder Printfox problemlos geladen und weiterverarbeitet werden kann. Zusätzlich Tips und Tricks für den C64, C128, C16 und Plus/4. Die Beschreibung finden Sie in Ausgabe 5/87
Diskette für C64

Bestell-Nr.: 10705 **DM 29,90*** sFr 24,90*/öS 299,–*

DFÜ: Mit dem C64 an das Telefonnetz

Proterm V6.0: Proterm V6.0 zeichnet sich durch seine großen Befehlsumfang und seine sehr leichte Bedienbarkeit aus. Dieses Terminalprogramm gestattet wirklich jedem einen bequemen Einstieg oder Aufstieg in die Welt der DFÜ. **Bonito-RCA-64:** Mit Hilfe dieses Programms und der dazugehörenden Hardware (Bauanleitung im Heft) können Sie mit Ihrem C64 Fernschreiben und Morsesendungen empfangen. **Fractale 9.0:** Auch Computer können kreativ sein. Ein gutes Beispiel dafür sind die Fractals, die richtig dargestellt, der Natur ähnliche Gebirgszüge und Landschaften zeigt. Mit diesem Programm können dies auch Sie auf Ihrem C64 verwirklichen. **Vizacalc:** Durch Vizacalc steht Ihnen von nun an im Texteditor von Vizawrite ein leistungsfähiger Taschenrechner und Kalender zur Verfügung. Zusätzlich sind auf der Diskette noch viele Tips und Trick für den C64/C128 und C16 enthalten. Die Beschreibungen finden Sie in Ausgabe 4/87
1 Diskette für den C64

Bestell-Nr.: 10704 **DM 29,90*** sFr 24,90*/öS 299,–*

Keine Chance für Killer-Tracks

Copy+: Copy+ ist ein hervorragendes Kopierprogramm. Alles, was Sie benötigen, ist ein leicht herzustellendes Parallelkabel zwischen Floppy und C64 . (Bauanleitung im Heft) Copy+ macht sogar vor »Read Errors« nicht halt, die andere Kopierprogramme in die Knie zwingen. **Arith13:** Arith13 ist eine Betriebssystemerweiterung die die Rechengenauigkeit Ihres C64 von 9 auf 13 Stellen erweitert und die Rechenfehler der Orginalroutinen ausschaltet. **Heureka:** Heureka erlaubt es Ihnen, Ihre Programme mit mehr als der 25fachen Geschwindigkeit, die Sie von Ihrer 1541 gewohnt waren, zu laden. **Sound-Cruncher:** Mit dem »Sound-Cruncher« können Sie Ihre Soundmonitor-Files auf einen Bruchteil der ursprünglichen Länge reduzieren. Weiterhin viele Tips und Tricks für den C64/C128 sowie C16. Die Beschreibungen finden Sie in Ausgabe 3/87
Diskette für C64

Bestell-Nr.: 10703 **DM 29,90*** sFr 24,90*/öS 299,–*

* Unverbindliche Preisempfehlung

Sie suchen packende Spiele, hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides! Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jedes Monat erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp. Wenn Sie Fragen zu den Programmen in unserem Angebot haben, rufen Sie uns an: Telefon (089) 4613-640 oder (089) 4613-133.

Bestellungen bitte an: Markt&Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656. Österreich: Ueberreuter Media Handels- und Verlagsgesellschaft mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0. Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar, und gegen Bezahlung einer Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die abgedruckte Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

Fantastische Hardcopies



Das Ausdrucken von Bildschirminhalten muß nicht schwierig sein. Mit dem richtigen Programm können Sie aus Ihrem Drucker oft Unglaubliches herausholen. Das geht sogar so weit, daß auch auf normalen Schwarzweiß-Druckern farbige Ausgabe möglich wird.

Das Programm »Galactical-Hardcopy« (Listing 1) kann Bilder in hochauflösender Grafik in 19 verschiedenen Formaten auf dem Epson-kompatiblen Drucker CP-80X ausdrucken. Das Programm wird dazu normal geladen und mit RUN gestartet. Danach erscheint ein Menübild und es wird das momentan eingestellte Druckformat angezeigt.

Laden von Bildern:

Zum Laden von Bildern muß die Taste <L> gedrückt werden. Bei der Eingabe des Namens kann man allerdings nicht die INST/DEL-Taste benutzen. Nach der Eingabe des Namens drücken Sie <RETURN>, und es erscheint die Frage, ob das gespeicherte Farb-RAM mitgeladen werden soll oder nicht. Wenn ja, dann antworten Sie mit <Y> für Ja, sonst mit <N> für Nein. Das Farb-RAM muß sich auf jeden Fall vor dem gespeicherten Grafikbild befinden, damit es miteingeladen werden kann. Danach muß die Diskette eingelegt und eine Taste gedrückt werden, damit das Bild in den Speicher des Computers geladen wird. Hat man sich dafür entschieden, kein Farb-RAM miteinzubinden, so erscheint das Bild in grüner Farbe vor schwarzem Hintergrund. Ein Tastendruck während des Ladens bricht den Ladevorgang ab.

Inhaltsverzeichnis:

Mit Taste <D> für Directory erhalten Sie das Inhaltsverzeichnis der Diskette aufgelistet. Die Ausgabe stoppt alle 15 Zeilen und kann auf Tastendruck fortgesetzt werden. Die Taste <RUN/STOP> bricht das Auflisten des Inhaltsverzeichnisses ab.

Grafik sehen:

Durch Drücken von <G> für Grafik wird vom Text- auf den Grafikbildschirm umgeschaltet. Drückt man erneut eine Taste, so wird wieder auf Textdarstellung umgeschaltet.

Invertieren von Bildern:

Ein Drücken der Taste <I> bewirkt ein Invertieren der gesamten Grafik und danach einen Einsprung in den Modus »Grafik sehen«.

Einstellen des Druckmodus:

Um den Druckmodus festzulegen, muß man die Taste <M> drücken. Danach kann gewählt werden, ob das Bild horizontal, vertikal oder als Briefkopf ausgedruckt werden soll. Horizontal bedeutet, daß das Bild, so wie es auf dem Bildschirm zu sehen ist, ausgegeben wird. Vertikal heißt, daß das Bild um 90 Grad gedreht auf dem Drucker ausgegeben wird. Briefkopf (letter-head) führt dazu, daß das Bild in zwei Hälften zerlegt nebeneinander gedruckt wird. In

Bildschirmzeilen ausgedrückt heißt das, daß links auf dem Blatt die Zeilen 1 bis 12 und rechts daneben die Zeilen 13 bis 24 ausgedruckt werden.



Bild 1. Mit den richtigen Hardcopy-Programmen für den Oki 20: In Farben schwelgen...

Einstellen des Druckformats:

Mit den Cursortasten kann man das Druckformat einstellen, wobei das eingestellte Format am Bildschirm angezeigt wird. Mit <CRSR-aufwärts> beziehungsweise <CRSR-abwärts> wird die Y-Breite und mit <CRSR-links> beziehungsweise <CRSR-rechts> wird die X-Breite eingestellt. Es gibt die Formate ½:½, ½:1, ½:2, 1:½, 1:1, 1:2, 2:½, 2:1 und 2:2 (X-Breite: Y-Breite). Das eingestellte Druckformat wird beim Ausdruck von Briefköpfen nicht berücksichtigt. Beim vertikalen Ausdruck der Bilder werden diese um 90 Grad nach rechts gedreht.

Einstellen des Seitenabstandes:

Um den Abstand des Bildes vom linken Rand einzustellen, müssen Sie die Taste <S> drücken und es erscheint in reverser Schrift »Set Margin« auf dem Bildschirm. Mit <CRSR-links> beziehungsweise <CRSR-rechts> kann der Wert verändert werden. Dieser Wert legt fest, um wie viele Leerzeichen das Bild beim Ausdruck nach rechts verschoben wird.

Man sollte jedoch darauf achten, daß der Wert nicht zu groß ist, da es sonst zu einem verzerrten Ausdruck des Bildes kommt. Bei einem Ausdruck in doppelter X-Breite sollte der Abstand zum Beispiel stets 0 sein. Hat man den gewünschten Wert eingestellt, gelangt man mit <RETURN> ins Menü zurück.

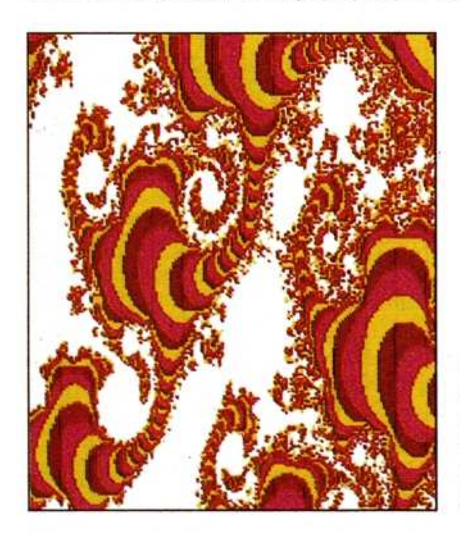


Bild 2. ...auch bei der mathematischen Faszination von Apfelmännchen (im hochgestellten Format)

Ausdrucken eines Bildes:

Damit das Bild in dem gewünschten Format ausgedruckt wird, drücken Sie <P> für »Print« und das Bild wird gedruckt. Der Ausdruck wird bei Betätigung einer anderen Taste abgebrochen.

Wichtiger Hinweis:

Damit das Programm mit anderen Epson-kompatiblen Druckern zusammenarbeiten kann, muß der Drucker folgende Steuercodes kennen:

ESC + "k" + n1 + n2: 640 Punkte pro Zeile (Anzahl der Grafikbytes = n1+256*n2)

ESC + "1" + n1 + n2: 1288 Punkte pro Zeile

ESC + "3" + n: Zeilenabstand = n/216 Zoll

(J. Clere/sk)

Hardcopy für Star SG-10

Dieses Programm (Listing 2) ist eine Anpassung der in Ausgabe 1/86 abgedruckten »Hardcopy auf Tastendruck« für den Star SG-10. Zur Initialisierung wird sie mit SYS 49152 aufgerufen. Die Hardcopy wird durch einmaligen Druck auf die Taste <F1> ausgelöst. Sollte sich das mit einer anderen Erweiterung überschneiden, können Sie in Speicherstelle 49168 einen anderen Wert, entsprechend des Tastaturcodes, POKEn. Zum Beispiel 51 für <HOME>. Eine Änderung der Sekundäradresse kann bei einigen Interfaces notwendig sein. Durch POKEn in Speicherstelle 49200 ist sie jederzeit zu ändern. (Dr. H. Schwartz/sk)



Bild 3. Auch Comics können jetzt farbig gestaltet werden mit dem Oki 20

Minihardcopy für Oki 20

Bitte geben Sie das Listing 3 mit dem MSE ein und speichern Sie es auf Diskette. Nun können Sie es jederzeit mit

```
LOAD"MINI MC-HARDCOPY",8,1
```

laden.

Legen Sie nun ein Farbband in Ihren Oki ein und schalten Sie ihn an.

Dann schieben Sie eine Diskette mit Grafiken im Koala-Painter-Format ins Laufwerk und rufen das Programm mit

```
SYS 49152,"BILD 1","BILD 2","BILD 3",lf
```

auf.

Für »BILD 1 bis 3« setzen Sie die Namen ein, die Sie Ihren Grafiken gegeben haben. Die Bilder werden dann nebeneinander in der Reihenfolge

Bild 1, ↑ Bild 2, ↑ Bild 3

ausgedruckt. Ein Beispiel sehen Sie in Bild 1 bis 3.

Wollen Sie weniger als drei Bilder drucken, dann geben Sie einfach für den entsprechenden Namen nur zwei Anführungszeichen ("") ein.

Der Parameter »LF« bestimmt die Größe des Zeilenvorschubs zwischen zwei Druckerzeilen. Ist »LF« ungleich 0, dann werden die durch den etwas ungenauen Papiertrans-



port verursachten weißen Striche unterdrückt, indem das Papier einfach um einen Punkt weniger vorgeschoben wird. Dadurch entstehen zwar manchmal dunkle Streifen, diese fallen jedoch weitaus weniger auf, als die hellen. Trotzdem kann es manchmal sinnvoll sein, mit »LF« gleich 0 zu drucken, besonders bei sehr hellen Bildern.

Hinweis: Das Programm ist entwickelt für einen Oki 20 mit integriertem C64-Interface. Über ein Centronics-Interface können sich erhebliche Probleme ergeben. Das Programm erwartet ferner die Bilder im Koala-Painter-Format auf Diskette. Wollen Sie also ein anderes Bild ausdrucken, dann müssen Sie dieses vorher ins Koala-Format übertragen. Dazu benutzen Sie am besten das Programm »Koala-Copy« aus dem 64'er Sonderheft Grafik, 8/86.

(S. Stuntz/sk)

Micro-Hardcopy mit Giga-CAD

Die Bedienung von »Giga-Hard« unterscheidet sich nicht wesentlich von der anderer Hardcopy-Routinen. Das Programm ist voll in Maschinensprache geschrieben und druckt auf einem Star NL-10 die Grafikseite ab $2000, sowie daneben die Grafikseite ab $A000. Dies ist dafür nötig, weil eine komplette Giga-CAD-Hardcopy aus drei nebeneinanderliegenden Bildern besteht, von denen jeweils eines im Speicher liegt (Bild 4).

Das heißt, daß zum Beispiel mit einem Monitor die (erste) linke Hälfte der Grafik in den Bereich ab $2000 zu laden ist und die (erste) rechte Hälfte in den Bereich von $A000.

Danach ist ein Kanal zum Drucker zu öffnen und die Routine zu starten mit

```
SYS ADRESSE # LFN
```

Die Adresse ist $CE00 (52736) bei der ersten Version (Listing 4). Allerdings gibt es auch eine weitere Version (Listing 5) ab $9000, damit noch irgendwo Platz für einen Monitor ist.

LFN ist die logische File-Nummer des Druckerkanals, auf dem die Daten ausgegeben werden.

Mit den weiteren Bildern der Giga-CAD-Grafik verfährt man analog. Allerdings ist sicherheitshalber nach jedem Grafik-Paar der Druckerkanal zu schließen.

Tips:

a) Mit POKE EINSPRUNGADRESSE + 222,A1H und POKE EINSPRUNGADRESSE + 224,A2H können die Bildschirmadressen geändert werden, falls $2000 und $A000 nicht zusagen. Dabei stellen »A1H« und »A2H« die beiden High-Bytes der entsprechenden Bildschirmadressen dar.

b) Um die originale Microhardcopy-Routine an andere Drucker anzupassen, muß ein Byte von 0 auf $0D (13) geändert werden, da in den definierten LFs offensichtlich kein CR enthalten ist.

Das entsprechende Byte der erweiterten Routine liegt bei STARTADRESSE+194.

Für Epson-Drucker muß man an diese Stelle eine 0 schreiben. Normalerweise steht hier wie gesagt $0D (13).

(L. Kellner/sk)

Farbig drucken

Wer farbige Bilder auf Papier bringen will, muß nicht unbedingt einen Farbdrucker besitzen. Farbfähige Drucker sind immer noch sehr teuer. Und wer nur gelegentlich in Farbe drucken will, mag dafür nicht gern tief in die Tasche greifen.

In Ausgabe 10/86 des 64'er-Magazins wurde bereits ein Programm veröffentlicht, das dem MPS 801 zur Farbe verhilft. »Softcolor« (Listing 6) macht den farbigen Druck nun auch für Epson-Drucker und Kompatible ohne großen Aufwand möglich. Der Farbdruck (Bild 5) macht deutlich, daß auf diese Weise mit geringem Aufwand durchaus ansprechende Bilder hergestellt werden können.

Der Farbdruck erfordert als zusätzliche »Hardware« nur entweder farbige Farbbänder oder aber farbiges Kohlepapier. Kohlepapier ist allerdings nur mit Einschränkungen

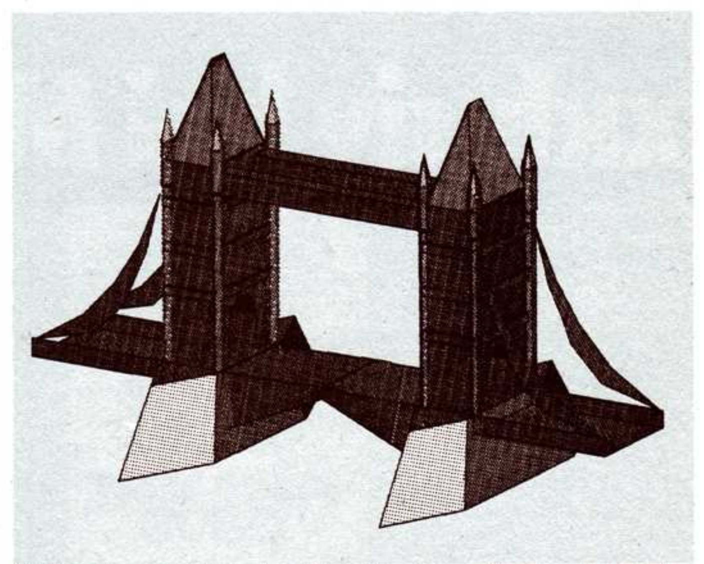


Bild 4. Das Ergebnis der Micro-Hardcopy für Giga-CAD. Es besteht aus mehreren nebeneinanderliegenden Bildern.

geeignet. Die Farben dieser Papiere sind meist sehr dunkel und haben wenig Leuchtkraft. Sie verschmieren, sind deshalb in der Handhabung riskant und auch die fertigen Bilder werden darum leicht verwischt. Außerdem bereitet das Positionieren des Papiers im Drucker ziemliche Schwierigkeiten, wenn Kohlepapier darüberliegt. Am ehesten noch gelingt dies, wenn man das Kohlepapier mit Klebeband vorsichtig auf dem Papier fixiert.

Wesentlich leichter wird die Arbeit mit verschiedenfarbigen Farbbändern. Deren Farben sind auch wesentlich ansprechender. Für die gängigsten Drucker sind farbige Farbbänder im Handel – wenn auch oft nur mit Mühe erhältlich. Die Mehrkosten im Vergleich zu Kohlepapier machen sich bald bezahlt.

Filtermethoden

Üblicherweise werden beim Farbdruck farbige Darstellungen dadurch erreicht, daß die gleichfarbigen Teile der Vorlage herausgefiltert und nach dem so gewonnenen Farbauszug eine einfarbige Druckplatte gefertigt wird. Das Bild der mehreren einfarbigen Druckplatten ergibt – exakt übereinandergedruckt – eine vielfarbige Darstellung der Vorlage, die nicht nur die Grundfarben, sondern auch Mischfarben enthält, die durch das Übereinanderdrucken mehrerer Farben entstehen. Die Farbtreue ist dabei um so höher, je besser die Farbauszüge die Farbdichteunterschiede der Vorlage wiedergeben. Auch beim Farbdruck mit dem Matrixdrucker ist es wünschenswert, nicht nur einfach rote, grüne oder blaue Flächen zu drucken, sondern die Farbsättigung der Flächen variieren und verschiedene Farben durch Übereinanderdrucken mischen zu können.

»Softcolor« stellt darum die Möglichkeit zur Verfügung, jeden Farbauszug in vier verschiedenen Stufen der Farbsättigung auszudrucken. Schon bei der Verwendung von nur zwei verschiedenen Farben ergibt sich damit eine große Vielfalt von möglichen Farbschattierungen und Mischfarben (Bild 5).

Das Programm besteht aus einem Maschinensprache-Programm (Listing 6), das die Druckroutine enthält. Die Parameter für den Ausdruck, vor allem die gewünschte Farbdichte für die verschiedenen Flächen der Grafik, müssen der Routine beim Aufruf übergeben werden.

Das Basic-Programm (Listing 7) dient dazu, diese Parameter für den Ausdruck zusammenzustellen und an die Druckroutine zu übergeben. Es bietet dazu eine einfache Menü-Steuerung und die wichtigsten Funktionen zum Laden einer Grafik und zur Steuerung des Drucks. Dabei ist vorausgesetzt, daß die auszudruckende Grafik bereits auf Diskette gespeichert ist. Die Grafik wird vom Programm aus in den Speicherbereich ab $2000 geladen und kann nun auf dem Bildschirm betrachtet und farblich verändert werden. Die jeweils »aktivierte« Bildschirmfarbe wird dabei blinkend dargestellt.

Nach dem Laden erscheint die Grafik zunächst in einer voreingestellten Farbkombination (sie kann geändert werden, indem in den Zeilen 260 und 270 jeweils die Ziffer 7 durch eine andere Ziffer zwischen 0 und 12 ersetzt wird). Beim Erscheinen der Grafik auf dem Bildschirm werden Sie bemerken, daß alle Flächen einer bestimmten Bildschirmfarbe blinken. Die Blinkfrequenz kann in Zeile 310 durch Verändern des Wertes 50 verstellt werden. Mit dem Blinken wird signalisiert, welche der vier Grundfarben der Multicolor-Grafik zur Zeit aktiviert ist. Die Bildschirmdarstellung dieser Farbe kann mit den Tasten <+> und <-> beliebig umgeschaltet werden.

Die nächste Grundfarbe wird aktiviert und zum Blinken gebracht durch Drücken der Taste <*>. Mit Hilfe dieser Funktionen kann auf dem Bildschirm die günstigste Farbgestaltung für den Ausdruck erprobt werden. (Auf die Farben der Hardcopy selbst hat aber die Farbgebung auf dem Bildschirm keinen Einfluß!)



Bild 5. Auf Epson-Druckern lassen sich auch schon mit zwei Farben interessante Farb-Varianten erzeugen

Zur Vorbereitung des Drucks müssen Sie zuerst die »Hardware« einrichten: Entweder wird dazu das Farbband mit der für den ersten Durchgang gewünschten Druckfarbe in den Drucker eingelegt, oder es ist über das Papier ein farbiges Kohlepapier zu legen. Außerdem ist unbedingt erforderlich, die Startposition des Papiers so zu markieren, daß es nach dem ersten Druckdurchgang wieder genau in diese Startposition gebracht werden kann.

Nun können Sie mit dem Druck beginnen. Mit der Taste <*> bringen Sie diejenigen Flächen der Grafik auf dem Bildschirm zum Blinken, die mit der Farbe ausgedruckt werden sollen, welche jetzt im Drucker liegt. Dann können Sie nach Drücken der Taste <£> auswählen, mit welcher Farbdichte die in der Grafik blinkenden Flächen gedruckt werden sollen. Wenn mit der aktuellen Druckfarbe nur diese Flächen gedruckt werden sollen, dann können Sie nun schon den Ausdruck des ersten Farbauszugs starten. Anderenfalls lassen Sie sich auf dem Bildschirm wieder die Grafik zeigen, wählen mit <*> andere Flächen an und bestimmen wieder nach Drücken der Taste <£>, mit welcher Farbdichte diese Flächen gedruckt werden sollen.

Nach dem Druck des ersten Farbauszugs muß das Farbband oder das Kohlepapier gewechselt und das Papier mit Hilfe der Startmarkierung wieder in die Anfangsposition gebracht werden. Danach kann der zweite und jeder weitere Farbauszug ebenso gewählt und ausgedruckt werden wie der erste.

Pixel für Pixel auf Farbe untersuchen

Kernstück des Programms ist die Druckroutine, die vom Basic-Programm aus in den Speicher ab $C000 nachgeladen wird. Diese Routine entspricht im wesentlichen dem im 64'er Sonderheft 4/85 auf Seite 56 veröffentlichten Hardcopy-Programm. Es ist für die hier gestellte Aufgabe verbessert und erweitert worden. Im Hinblick auf diese frühere Veröffentlichung können hier die Erläuterungen zur Funktionsweise knapp gehalten werden:

Die Druckroutine untersucht bei jedem Pixel des Bildschirms, in welcher Bildschirmfarbe er gesetzt ist, ob er also in einer der beiden Hintergrund- oder einer der beiden Zeichenfarben darzustellen ist. Anschließend wird geprüft, ob und mit welcher Dichte diese Farbe ausgedruckt werden soll. Diese Information wird mit den letzten vier Parametern des SYS-Aufrufs (Zeile 700) übergeben. Von diesen vier Parametern bestimmt der erste <F1> die Farbdichte der ersten Bildschirmfarbe, der zweite <F2> die Dichte der zweiten Farbe und so weiter.

Dabei signalisiert jeweils der Variablenwert 0 den Ausdruck mit größtmöglicher Dichte. Höhere Werte führen zum Ausdruck mit geringerer Dichte. Werte über 3 bewirken, daß die Bildschirmflächen mit der entsprechenden Farbe gar nicht ausgedruckt werden.

In der hier abgedruckten Version ist der SYS-Aufruf in Zeile 700 so gefaßt, daß der Ausdruck über den seriellen Bus und ein Hardware-Interface erfolgen kann. Es besteht aber auch die Möglichkeit, ohne eine solche Schnittstelle auszukommen. Wenn Sie in Zeile 700 den OPEN- und den CLOSE-Befehl löschen und als zweiten Parameter des SYS-Aufrufs statt der 4 eine 0 setzen, dann wird der Drucker mit einem Centronics-Kabel unmittelbar über den User-Port angesprochen.

Bleibt noch zu klären, welche Grafiken sich für den Farbdruck mit Softcolor eignen. Im Prinzip kann jede Grafik farbig dargestellt werden. Allerdings wird die Art der Farbcodierung manchmal zu wenig attraktiven Ergebnissen führen. Gute Farbdrucke können Sie aber mit Sicherheit von Hi-Eddi-, Simons-Basic- und Apfelmännchen-Bildern erwarten, also Grafiken im eingeschränkten Farbmodus. Im Zweifelsfalle hilft nur Ausprobieren.

Grafiken aus fremden Programmen

Sie werden bestimmt viele geeignete Farbbilder finden, wenn Sie sich einmal die Grafiken in Ihrer Spielesammlung anschauen. Probleme kann es allerdings bereiten, diese Bilder als Grafik-File zu isolieren und auf Diskette zu speichern. Darum noch ein Hinweis auf das Programm »Hardmaker« in dieser Ausgabe, das diese Aufgabe beispielsweise für Sie erledigt.

(J. Curdt/sk)



GEOS für den C128 (englisch)

Der neue Betriebssystemstandard – in der Originalversion für den C128. GEOS 64 wurde an den 128er-Modus des C128 angepaßt und kann sowohl die doppelte Auflösung als auch den größeren Speicher nutzen. Unterstützt werden am RGB-Eingang angeschlossene Monitore (80 Zeichen), sowie die üblichen PAL-Monitore und Fernsehapparate. Ansonsten gelten die Leistungsmerkmale von GEOS 64.

Hardware-Anforderung:
C128, Floppy 1541, 1570 oder 1571, Joystick oder Maus 1531.
5¼-Zoll-Diskette
Bestell-Nr. 50328 **DM 119,–***

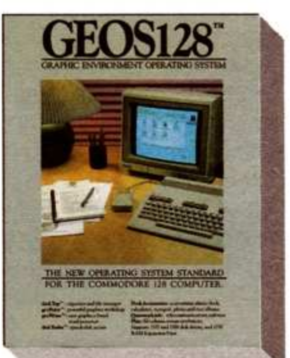
GEOS für den C128 (deutsch)

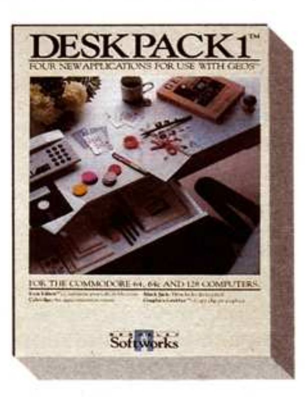
Bestell-Nr. 50327 **DM 119,–***

Deskpack 1/GeoDex für den C64/C128 (deutsch)

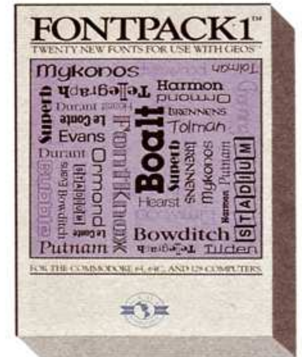
Deskpack 1/GeoDex: die nützlichen Zusatzprogramme für GEOS Graphics-Grabber! Überträgt Grafiken von Print Shop, Print Master und Newsroom zur Anwendung mit GeoPaint und GeoWrite. Leistungsumfang: Icon Editor – erstellt und verändert Icons nach Ihren Vorstellungen. GeoDex – Adreß- und Notizbuch mit Modemunterstützung. GeoMerge – Suchen nach Adreßgruppen aus GeoDex sowie Erstellen von Formbriefen und Listen. Blackjack – das klassische Glücksspiel. Kalender.

Hardware-Anforderungen:
C64 oder C128, Floppy 1541, 1570 oder 1571, Joystick.
Software-Anforderung: GEOS 64.
Bestell-Nr. 50322 **DM 69,–***









GEOS, Version 1.3, für den C64/C128 (deutsch)

Der neue Betriebssystemstandard für Commodore 64. Leistungsumfang: DeskTop – das Grafikinterface zum GEOS-Betriebssystem. Schauen Sie sich die Dateien als Icons oder im Textmodus an. Automatisches Sortieren von Dateien nach Alphabet, Größe, Typ oder Datum der letzten Änderung ist kein Problem. Dateien kopieren, löschen und Disketten formatieren ist natürlich enthalten. GeoPaint: ein umfangreiches Zeichenprogramm in Farbe mit 14 verschiedenen Grafiktools, 32 Pinselstärken, 32 verschiedenen Mustern. GeoWrite: ein einfaches, leichtbedienbares Textprogramm. Desk-Accessories: Wecker, Notizblock, Taschenrechner.

Hardware-Anforderungen:
C64 oder C128 (64er-Modus), Floppy 1541, 1570 oder 1571, Joystick.
Bestell-Nr. 50320 **DM 59,–***

Update von älteren englischen Versionen auf die neue deutsche Version 1.3. Erhältlich direkt beim Markt&Technik-Buchverlag gegen Einsendung des Originalprodukts und gegen Vorauskasse.
Bestell-Nr. 50320U **DM 39,–***

Fontpack 1 für den C64/C128 (deutsch)

Die unentbehrliche Utility für GEOS-Benutzer! Fontpack 1 wurde für die GEOS-Applikationen GeoPaint und GeoWrite entwickelt und enthält 20 neue, außergewöhnliche Schriftarten, die jeden Anwender begeistern werden.

Hardware-Anforderungen:
C64 oder C128, Floppy 1541, 1570 oder 1571, Joystick.
Software-Anforderungen: GEOS 64
Bestell-Nr. 50321 **DM 49,–***

GeoWrite Workshop für den C64/C128

Bestell-Nr. 50323 **DM 89,–***

GeoFile für den C64/C128

Bestell-Nr. 50324 **DM 89,–***

GeoCalc für den C64/C128

Bestell-Nr. 50325 **DM 89,–***

* Unverbindliche Preisempfehlung

In Vorbereitung:

GeoWrite Workshop 128
Bestell-Nr. 50329 ca. **DM 119,–***

GeoFile 128
Bestell-Nr. 50330 ca. **DM 119,–***

GeoCalc 128
Bestell-Nr. 50331 ca. **DM 119,–***



Zeitschriften · Bücher
Software · Schulung

Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften oder in den Fachabteilungen der Warenhäuser.

711241

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.

```
Name : g-hc                      0801 166c
-----------------------------------------
0801 : 28 08 0a 00 9e 20 32 30   c4
0809 : 39 32 20 3a 8f 14 14 14   bd
0811 : 14 20 28 57 29 20 4a 4f   86
0819 : 45 52 47 20 43 4c 45 56   b6
0821 : 45 20 31 39 38 36 00 00   1f
0829 : 00 00 00 a9 00 8d 20 d0   ed
0831 : 8d 21 d0 4c 05 12 a9 00   94
0839 : 8d 43 08 a9 60 8d 44 08   33
0841 : 60 ad cc cc 60 bd cc cc   05
0849 : 60 bd cc cc 60 ea a9 0c   71
0851 : 8d 4e 08 a9 a0 8d 87 09   e3
0859 : 20 37 08 20 15 09 20 01   37
0861 : 09 ad 43 08 8d 47 08 ad   a1
0869 : 44 08 8d 48 08 20 a4 08   42
0871 : 18 ad 43 08 69 00 8d 47   8d
0879 : 08 ad 44 08 69 0f 8d 48   40
0881 : 08 20 a4 08 18 ad 43 08   d0
0889 : 69 40 8d 43 08 ad 44 08   ed
0891 : 69 01 8d 44 08 ce 4e 08   a7
0899 : d0 c7 20 cc ff a9 04 20   8c
08a1 : c3 ff 60 20 4f 09 a9 28   b4
08a9 : 8d ec 08 a2 07 20 46 08   9d
08b1 : 9d ed 08 ca 10 f7 a0 07   f2
08b9 : ce 86 09 10 06 20 4f 09   bf
08c1 : ce 86 09 a2 07 a9 00 1e   63
08c9 : ed 08 6a ca 10 f9 20 e9   d3
08d1 : 0e 88 10 ef 18 ad 47 08   41
08d9 : 69 08 8d 47 08 ad 48 08   b2
08e1 : 69 00 8d 48 08 ce ec 08   71
08e9 : d0 c1 60 ea 00 00 00 00   0f
08f1 : 00 00 00 00 00 00 00 00   f2
08f9 : 00 00 00 00 00 00 00 00   fa
0901 : a9 1b 20 d2 ff a9 33 20   f5
0909 : d2 ff a9 16 20 d2 ff a9   f4
0911 : 0d 4c d2 ff a9 04 a2 04   46
0919 : a0 01 20 ba ff a9 00 20   27
0921 : bd ff 20 c0 ff a2 04 20   63
0929 : c9 ff b0 01 60 a2 28 20   3a
0931 : e3 12 20 2d 0e 20 e4 ff   41
0939 : f0 fb a2 00 20 e3 12 a2   7e
0941 : 02 20 e3 12 68 68 68 68   cb
0949 : 20 e7 ff 4c 7e 11 a5 c6   7b
0951 : d0 35 a9 1b 20 d2 ff ad   7e
0959 : 00 12 c9 01 d0 0f ad 03   37
0961 : 12 c9 02 f0 08 a9 4c 20   36
0969 : d2 ff 4c 73 09 a9 4b 20   08
0971 : d2 ff ad 87 09 4a 4a 4a   40
0979 : 8d 86 09 a9 a0 20 d2 ff   17
0981 : a9 00 4c d2 ff ea ea 68   6b
0989 : 68 68 68 a9 00 85 c6 20   fc
0991 : cc ff a9 04 4c c3 ff 48   bb
0999 : 20 9e 09 68 60 a5 c6 d0   48
09a1 : 2f a9 1b 20 d2 ff ad 01   56
09a9 : 12 c9 01 d0 0f ad 03 12   89
09b1 : c9 02 f0 08 a9 4c 20 d2   dc
09b9 : ff 4c c2 09 a9 4b 20 d2   cb
09c1 : ff a9 64 8d 86 09 a9 c8   48
09c9 : 20 d2 ff a9 00 4c d2 ff   35
09d1 : 68 68 68 4c 88 09 20 c1   e6
09d9 : 0e 20 98 09 a9 19 8d ec   b2
09e1 : 08 a2 07 20 46 08 20 18   56
09e9 : 0f ca 10 f7 38 ad 47 08   7e
09f1 : e9 40 8d 47 08 ad 48 08   66
09f9 : e9 01 8d 48 08 ce ec 08   8a
0a01 : d0 df 60 a9 00 8d 43 08   98
0a09 : a9 7e 8d 44 08 60 a2 00   eb
0a11 : 20 e3 12 a2 04 20 e3 12   f1
0a19 : 20 57 0b a9 01 a2 c9 a0   6a
0a21 : 0a 20 bd ff a9 02 a2 08   f0
0a29 : a0 00 20 ba ff 20 c0 ff   2d
0a31 : 20 82 0b a2 02 20 c6 ff   e6
0a39 : 20 e4 ff 20 e4 ff a2 08   b8
0a41 : a0 06 8e cb 0a 8c cc 0a   4e
0a49 : 18 ae cb 0a ac cc 0a 20   86
0a51 : f0 ff 20 e4 ff 20 e4 ff   7a
0a59 : c9 01 d0 57 20 e4 ff 8d   06
0a61 : 52 0b 20 e4 ff 8d 53 0b   ad
0a69 : 20 cd 0a a9 23 8d ca 0a   05
0a71 : 20 e4 ff f0 0c 20 d2 ff   2e
0a79 : ce ca 0a d0 f3 20 82 0b   a9
0a81 : 60 a9 92 20 d2 ff ee cb   df
0a89 : 0a ad cb 0a c9 18 d0 b8   b0
0a91 : 20 cc ff 20 e4 ff f0 fb   25
0a99 : c9 03 f0 12 a2 00 20 e3   d5
0aa1 : 12 a2 04 20 e3 12 a2 02   67
0aa9 : 20 c6 ff 4c 3f 0a a9 07   af
0ab1 : 8d cb 0a 20 cc ff a9 02   22
0ab9 : 20 c3 ff ad cb 0a c9 07   b3
0ac1 : f0 05 20 e4 ff f0 fb 60   11
0ac9 : 24 ea ea ea a0 00 8c 51   59
0ad1 : 0b a9 10 8d 54 0b a9 27   f9
0ad9 : 8d 55 0b 20 0f 0b a9 e8   99
0ae1 : 8d 54 0b a9 03 8d 55 0b   98
0ae9 : 20 0f 0b a9 64 8d 54 0b   a3
0af1 : a9 00 8d 55 0b 20 0f 0b   ac
0af9 : a9 0a 8d 54 0b a9 00 8d   ae
0b01 : 55 0b 20 0f 0b 18 ad 52   93
0b09 : 0b 69 30 4c d2 ff a9 2f   91
0b11 : 8d 56 0b ee 56 0b 38 ad   64
0b19 : 52 0b ed 54 0b 8d 52 0b   73
0b21 : ad 53 0b ed 55 0b 8d 53   83
0b29 : 0b b0 e8 ad 52 0b 6d 54   58
0b31 : 0b 8d 52 0b ad 53 0b 6d   75
0b39 : 55 0b 8d 53 0b ad 56 0b   6f
0b41 : c9 30 d0 06 ae 51 0b d0   5a
0b49 : 01 60 ee 51 0b 4c d2 ff   be
0b51 : ea ea ea ea ea ea a9 00   75
0b59 : 20 bd ff a9 0f a2 08 a0   f4
0b61 : 0f 20 ba ff 20 c0 ff a2   7c
0b69 : 0f 20 c9 ff b0 03 4c cc   e9
0b71 : ff 20 e7 ff a2 06 20 e3   1d
0b79 : 12 20 e4 ff f0 fb 68 68   36
0b81 : 60 20 cc ff a2 0f 20 c6   d5
0b89 : ff 20 e4 ff c9 30 d0 03   39
0b91 : 4c cc ff 20 d2 ff 20 e4   bf
0b99 : ff c9 0d d0 f6 20 d2 ff   96
0ba1 : a9 0f 20 c3 ff 20 e7 ff   f3
0ba9 : 20 e4 ff f0 fb 68 68 60   bf
0bb1 : ea 31 32 33 34 35 36 37   5b
0bb9 : 38 39 30 31 32 33 34 35   b8
0bc1 : 36 ea ea a2 00 20 e3 12   30
0bc9 : a2 22 20 e3 12 a2 0f a9   c7
0bd1 : 20 9d b2 0b ca 10 fa e8   b9
0bd9 : 8e b1 0b a9 12 20 d2 ff   a5
0be1 : ae b1 0b bd b2 0b 20 d2   8c
0be9 : ff a9 92 20 d2 ff a9 9d   74
0bf1 : 20 d2 ff 20 e4 ff f0 fb   88
0bf9 : c9 0d f0 5d c9 1d f0 35   e4
0c01 : c9 9d f0 3a c9 03 f0 2c   ed
0c09 : c9 22 f0 e7 c9 20 90 e3   c4
0c11 : c9 80 90 04 c9 a0 90 db   5b
0c19 : ae b1 0b 9d b2 0b 20 d2   c0
0c21 : ff ee b1 0b e8 e0 10 d0   dd
0c29 : b2 ce b1 0b a9 9d 20 d2   be
0c31 : ff 4c dc 0b 60 ae b1 0b   47
0c39 : bd b2 0b 4c 19 0c ae b1   ac
0c41 : 0b f0 b0 bd b2 0b 20 d2   52
0c49 : ff a9 9d 20 d2 ff a9 9d   97
0c51 : 20 d2 ff ce b1 0b 4c dc   12
0c59 : 0b ae b1 0b bd b2 0b 20   67
0c61 : d2 ff a2 0f bd b2 0b c9   ee
0c69 : 20 d0 04 ca 10 f6 60 e8   58
0c71 : 8e c2 0b a2 26 20 e3 12   8f
0c79 : 20 e4 ff c9 59 f0 04 c9   05
0c81 : 4e d0 f5 8d c3 0b 20 d2   21
0c89 : ff a2 24 20 e3 12 20 e4   00
0c91 : ff f0 fb 20 57 0b a9 01   82
0c99 : a2 08 a0 00 20 ba ff ad   9b
0ca1 : c2 0b a2 b2 a0 0b 20 bd   46
0ca9 : ff 20 c0 ff 20 82 0b 20   6b
0cb1 : 12 0e a9 00 8d fb 0d 8d   3d
0cb9 : ca 0c a9 5c 8d fc 0d 8d   8f
0cc1 : cb 0c a0 04 a9 50 a2 00   e3
0cc9 : 9d cc cc ca d0 fa ee cb   91
0cd1 : 0c 88 d0 f4 a9 00 8d 07   d3
0cd9 : 0e 8d ec 0c a9 60 8d 08   4e
0ce1 : 0e 8d ed 0c a0 20 a2 00   48
0ce9 : a9 00 bd cc cc ca d0 fa   f8
0cf1 : ee ed 0c 88 d0 f4 a2 01   2b
0cf9 : 20 c6 ff 20 e4 ff 20 e4   19
0d01 : ff ad c3 0b c9 4e f0 1e   38
0d09 : a9 04 8d 3d 0e a9 00 8d   09
0d11 : 3e 0e 20 e4 ff 20 fa 0d   02
0d19 : a5 c6 d0 28 ce 3e 0e d0   13
0d21 : f1 ce 3d 0e d0 ec a9 20   e6
0d29 : 8d 3d 0e a9 41 8d 3e 0e   a3
0d31 : 20 e4 ff 20 06 0e a5 c6   bc
0d39 : d0 0a ce 3e 0e d0 f1 ce   57
0d41 : 3d 0e d0 ec a9 00 85 c6   95
0d49 : 20 cc ff 20 e0 0d a9 01   f2
0d51 : 20 c3 ff a9 0f 20 c3 ff   89
0d59 : 20 e4 ff f0 fb 4c 2d 0e   fc
0d61 : 20 c1 0e 20 98 09 a9 19   94
0d69 : 8d ec 08 a2 07 8e b5 0d   99
0d71 : 18 ad 47 08 69 08 8d 4b   d6
0d79 : 08 ad 48 08 69 00 8d 4c   d0
0d81 : 08 ae b5 0d 20 46 08 8d   5f
0d89 : ff 10 20 4a 08 8d 00 11   f1
0d91 : 20 dc 10 20 18 0f ce b5   c8
0d99 : 0d 10 e6 38 ad 47 08 e9   78
0da1 : 40 8d 47 08 ad 48 08 e9   8c
0da9 : 01 8d 48 08 ce ec 08 d0   9a
0db1 : ba 4c dc 0e ea 20 06 10   72
0db9 : a9 00 8d a8 0f 20 b0 10   b0
0dc1 : 20 61 0d a9 01 8d a8 0f   48
0dc9 : 20 c6 10 20 61 0d 20 14   7c
0dd1 : 10 20 14 10 d0 e2 20 cc   37
0dd9 : ff a9 04 20 c3 ff 60 a9   c3
0de1 : 40 8d 07 0e a9 7f 8d 08   48
0de9 : 0e a2 02 a0 00 a9 00 20   6b
0df1 : 06 0e 88 d0 fa ca d0 f7   74
0df9 : 60 8d cc cc ee fb 0d d0   91
0e01 : 03 ee fc 0d 60 8d cc cc   9b
0e09 : ee 07 0e d0 03 ee 08 0e   fc
0e11 : 60 ad 00 dd 29 fc 09 02   a6
0e19 : 8d 00 dd ad 18 d0 a9 78   73
0e21 : 8d 18 d0 ad 11 d0 a9 3b   59
0e29 : 8d 11 d0 60 a9 17 8d 18   39
0e31 : d0 a9 1b 8d 11 d0 a9 c7   1c
0e39 : 8d 00 dd 60 ea ea a9 00   f6
0e41 : 8d 54 0e 8d 59 0e a9 60   9b
0e49 : 8d 55 0e 8d 5a 0e a0 20   8f
0e51 : a2 00 bd cc cc 49 ff 9d   4e
0e59 : cc cc ca d0 f5 ee 55 0e   a0
0e61 : ee 5a 0e 88 d0 ec 60 20   47
0e69 : 12 0e ad 03 12 f0 08 c9   aa
0e71 : 01 f0 14 20 4f 08 60 ac   04
0e79 : 01 12 c0 01 f0 22 c0 02   fb
0e81 : f0 24 c0 03 f0 14 60 ae   a2
0e89 : 00 12 e0 01 f0 15 e0 02   2a
0e91 : f0 05 e0 03 f0 07 60 4c   fe
0e99 : 01 0f 4c a9 0f 4c 29 10   82
0ea1 : 4c 86 10 4c b6 0d 20 ca   a8
0ea9 : 0f 20 c1 0e 20 a4 08 20   82
0eb1 : e6 0f 20 f1 0f d0 f2 20   e9
0eb9 : cc ff a9 04 20 c3 ff 60   51
0ec1 : ae 04 12 f0 09 a9 20 20   b3
0ec9 : d2 ff ca 4c c4 0e ad 43   d1
0ed1 : 08 8d 47 08 ad 44 08 8d   ab
0ed9 : 48 08 60 ad 00 12 c9 03   b1
0ee1 : f0 05 a9 0d 4c d2 ff 60   7c
0ee9 : 48 ad 03 12 c9 02 f0 0c   93
0ef1 : ad 00 12 c9 03 d0 05 68   f8
0ef9 : 48 20 d2 ff 68 4c d2 ff   3a
0f01 : 20 06 10 20 d7 09 20 dc   2c
0f09 : 0e 20 14 10 d0 f5 20 cc   05
0f11 : ff a9 04 20 c3 ff 60 ac   01
0f19 : 01 12 c0 03 f0 03 4c d2   b2
0f21 : ff ac 86 09 d0 05 48 20   d0
0f29 : 9e 09 68 ce 86 09 48 20   52
0f31 : d2 ff 68 4c d2 ff 20 c1   d8
0f39 : 0e 20 4f 09 a9 28 8d ec   38
0f41 : 08 a2 07 20 46 08 9d ed   57
0f49 : 08 ca 10 f7 a0 07 ce 86   44
0f51 : 09 10 06 20 4f 09 ce 86   6d
0f59 : 09 a2 07 a9 00 1e ed 08   63
0f61 : 6a ca 10 f9 20 87 0f 20   2e
0f69 : e9 0e 88 10 ec 18 ad 47   52
0f71 : 08 69 08 8d 47 08 ad 48   de
0f79 : 08 69 00 8d 48 08 ce ec   c1
0f81 : 08 d0 be 4c e6 0f ae a8   1d
0f89 : 0f d0 04 0a 0a 0a 0a a2   a1
0f91 : 00 8e a7 0f a2 04 0a 08   27
0f99 : 2e a7 0f 28 2e a7 0f ca   56
0fa1 : d0 f4 ad a7 0f 60 ea ea   c1
0fa9 : 20 ca 0f a9 01 8d a8 0f   65
0fb1 : 20 37 0f a9 00 8d a8 0f   93
0fb9 : 20 37 0f 20 f1 0f d0 eb   ef
0fc1 : 20 cc ff a9 04 20 c3 ff   cd
0fc9 : 60 a9 a0 ae 00 12 e0 03   16
0fd1 : d0 02 a9 50 8d 87 09 a9   a3
0fd9 : 19 8d 4e 08 20 37 08 20   6a
0fe1 : 15 09 4c 01 09 ad 00 12   d0
0fe9 : c9 03 f0 03 20 dc 0e 60   b2
0ff1 : 18 ad 43 08 69 40 8d 43   07
0ff9 : 08 ad 44 08 69 01 8d 44   47
1001 : 08 ce 4e 08 60 a9 28 8d   14
1009 : 4e 08 20 04 0a 20 15 09   ec
1011 : 4c 01 09 18 ad 43 08 69   0b
1019 : 08 8d 43 08 ad 44 08 69   aa
1021 : 00 8d 44 08 ce 4e 08 60   3a
1029 : 20 06 10 a9 01 8d a8 0f   c3
1031 : 20 49 10 a9 00 8d a8 0f   5c
1039 : 20 49 10 20 14 10 d0 eb   e3
1041 : 20 cc ff a9 04 4c c3 ff   ae
1049 : 20 c1 0e 20 98 09 a9 19   7c
1051 : 8d ec 08 a2 07 8e 85 10   c6
1059 : ae 85 10 20 46 08 20 87   06
1061 : 0f 20 18 0f ce 85 10 10   e2
1069 : ef 38 ad 47 08 e9 40 8d   b5
1071 : 47 08 ad 48 08 e9 01 8d   20
1079 : 48 08 ce ec 08 d0 d4 a9   c4
1081 : 0d 4c d2 ff ea 20 ca 0f   62
1089 : 20 b0 10 a9 00 8d a8 0f   68
```

Listing 1. »Galactical-Hardcopy« geben Sie bitte mit dem MSE (Seite 159) ein

```
1091 : 20 01 11 20 c6 10 a9 01  10
1099 : 8d a8 0f 20 01 11 20 f1  3f
10a1 : 0f 20 f1 0f 10 e2 20 cc  51
10a9 : ff a9 04 20 c3 ff 60 ad  9b
10b1 : 00 12 c9 03 f0 16 a9 1b  2a
10b9 : 20 d2 ff a9 33 20 d2 ff  f7
10c1 : a9 01 4c d2 ff ad 00 12  ea
10c9 : c9 03 f0 ea a9 1b 20 d2  47
10d1 : ff a9 33 20 d2 ff a9 15  74
10d9 : 4c d2 ff ae a8 0f d0 06  b6
10e1 : 4e ff 10 6e 00 11 a2 08  24
10e9 : 4e ff 10 6e 00 11 6e a7  9a
10f1 : 0f 4e ff 10 6e 00 11 ca  ea
10f9 : d0 ee ad a7 0f 60 ea ea  16
1101 : 20 c1 0e 20 4f 09 a9 28  be
1109 : 8d ec 08 a2 07 20 46 08  fd
1111 : 9d ed 08 ca 10 f7 a2 07  5a
1119 : 18 ad 47 08 69 40 8d 4b  40
1121 : 08 ad 48 08 69 01 8d 4c  80
1129 : 08 20 4a 08 9d f9 08 ca  34
1131 : 10 f7 a0 07 ce 86 09 10  ab
1139 : 06 20 4f 09 ce 86 09 a2  cf
1141 : 07 a9 00 1e ed 08 6a ca  3f
1149 : 10 f9 8d ff 10 a2 07 a9  3f
1151 : 00 1e f9 08 6a ca 10 f9  11
1159 : 8d 00 11 20 dc 10 20 e9  d1
1161 : 0e 88 10 db 18 ad 47 08  4f
1169 : 69 08 8d 47 08 ad 48 08  42
1171 : 69 00 8d 48 08 ce ec 08  01
1179 : d0 91 4c e6 0f a9 06 ac  b1
1181 : 01 12 88 f0 06 18 69 06  9e
1189 : 4c 83 11 8d 02 12 ad 00  f4
1191 : 12 0a 18 6d 02 12 aa 20  f8
1199 : e3 12 ad 03 12 0a 18 69  f6
11a1 : 1a aa 20 e3 12 18 a2 0e  1d
11a9 : a0 0f 20 f0 ff a9 20 20  05
11b1 : d2 ff 18 a2 0e a0 0e 20  3c
11b9 : f0 ff a9 00 8d 53 0b ad  0e
11c1 : 04 12 8d 52 0b 4c cd 0a  da
11c9 : a2 20 20 e3 12 20 e4 ff  b6
11d1 : f0 fb c9 9d f0 0b c9 1d  ae
11d9 : f0 15 c9 0d d0 ef 4c 1a  5a
11e1 : 12 ad 04 12 f0 e7 ce 04  9f
11e9 : 12 20 7e 11 4c ce 11 ad  a8
11f1 : 04 12 c9 4f f0 d7 ee 04  ec
11f9 : 12 20 7e 11 4c ce 11 ea  32
1201 : ea ea ea ea a9 02 8d 00  59
1209 : 12 8d 01 12 a9 00 8d 03  3b
1211 : 12 8d 04 12 a2 00 20 e3  a0
1219 : 12 a2 02 20 e3 12 20 7e  4d
1221 : 11 20 e4 ff f0 fb c9 9d  cd
1229 : f0 43 c9 1d f0 32 c9 91  bc
1231 : f0 55 c9 11 f0 44 c9 4c  51
1239 : f0 6f c9 51 f0 1b c9 44  15
1241 : f0 18 c9 4d f0 4e c9 49  95
1249 : f0 6e c9 50 f0 70 c9 47  35
1251 : f0 5d c9 53 f0 50 4c 22  d4
1259 : 12 60 20 0f 0a 4c 15 12  01
1261 : ad 00 12 c9 03 f0 ba ee  4d
1269 : 00 12 4c 1f 12 ad 00 12  1c
1271 : c9 01 f0 ad ce 00 12 4c  7a
1279 : 1f 12 ad 01 12 c9 03 f0  8a
1281 : a0 ee 01 12 4c 1f 12 ad  7c
1289 : 01 12 c9 01 f0 93 ce 01  0f
1291 : 12 4c 1f 12 ad 03 12 c9  a2
1299 : 02 d0 05 a9 ff 8d 03 12  16
12a1 : ee 03 12 4c 1f 12 4c c9  66
12a9 : 11 20 c4 0b 4c 15 12 20  53
12b1 : 12 0e 20 59 0d 4c 22 12  dd
12b9 : 20 3f 0e 4c b0 12 20 68  73
12c1 : 0e 20 2d 0e 4c 22 12 a9  5e
12c9 : 1b 20 d2 ff a9 32 20 d2  fb
12d1 : ff a9 0d 20 d2 ff 20 cc  33
12d9 : ff a9 04 20 c3 ff 4c 22  64
12e1 : 12 ea bd 0f 13 8d 3a 13  66
12e9 : bd 10 13 8d 3b 13 a2 00  fc
12f1 : 20 39 13 f0 18 c9 01 d0  06
12f9 : 0e 20 39 13 a8 20 39 13  5f
1301 : 18 20 f0 ff 4c f1 12 20  42
1309 : d2 ff 4c f1 12 60 47 13  94
1311 : 09 14 93 14 a1 14 bf 14  6e
1319 : dc 14 f9 14 16 15 33 15  01
1321 : 50 15 6d 15 8a 15 a7 15  14
1329 : c4 15 d3 15 e2 15 f1 15  d8
1331 : fd 15 1a 16 2d 16 4b 16  df
1339 : ae cc cc 18 ee 3a 13 d0  32
1341 : 03 ee 3b 13 8a 60 93 08  f7
1349 : 0e 1e b0 ae 20 20 20 b0  4d
1351 : 20 20 20 20 20 20 ac 20  83
1359 : 20 20 20 20 b0 20 20 20  62
1361 : ae b0 20 20 20 20 20 20  37
1369 : b0 0d dd ae b0 ae 20 dd  aa
1371 : b0 ae b0 20 db 20 ae b0  83
1379 : 20 b0 ae 20 dd 20 c0 20  c3
```

```
1381 : ab b3 b0 ae 20 ab ae b0  83
1389 : b3 b0 20 b0 ae b0 ae ae  3b
1391 : b0 0d ad bd ad b1 20 bd  4f
1399 : ad b1 ad bd ad 20 ad ad  30
13a1 : bd ad b1 20 bd 20 20 a0  44
13a9 : bd ad ad b1 20 bd 20 ad  aa
13b1 : b1 ad bd ad bd ab bd ad  e9
13b9 : b3 0d 20 20 20 20 20 20  c3
13c1 : 20 20 20 20 20 20 20 20  c1
13c9 : 20 20 20 20 20 20 20 20  c9
13d1 : 20 20 20 20 20 20 20 20  d1
13d9 : 20 20 20 20 20 bd 20 ad  e1
13e1 : bd 0d 20 20 20 20 20 20  f5
13e9 : 28 d7 29 20 20 20 20 42  53
13f1 : 59 20 ca 4f 45 52 47 20  3b
13f9 : c3 4c 45 56 45 20 49 4e  16
1401 : 20 31 39 38 36 11 0d 00  2f
1409 : 01 0e 07 d0 52 49 4e 54  3f
1411 : 4d 45 4e 55 20 3a 0d 01  49
1419 : 05 09 c6 4f 52 4d 41 54  7c
1421 : 20 3a 01 05 0d cd 4f 44  44
1429 : 55 53 20 20 3a 01 05 0e  10
1431 : cd 41 52 47 49 4e 20 3a  18
1439 : 01 04 10 28 c4 29 49 52  a5
1441 : 45 43 54 4f 52 59 01 1b  51
1449 : 10 28 cc 29 4f 41 44 01  d8
1451 : 06 11 28 c7 29 52 41 46  9a
1459 : 49 43 01 1a 11 28 c9 29  93
1461 : 4e 56 45 52 54 01 06 12  00
1469 : 28 d0 29 52 49 4e 54 01  e8
1471 : 1a 12 28 cd 29 4f 44 55  21
1479 : 53 01 0e 14 28 d3 29 45  a3
1481 : 54 20 cd 41 52 47 49 4e  a2
1489 : 01 11 16 28 d1 29 55 49  ec
1491 : 54 00 01 0f 06 c4 49 52  58
1499 : 45 43 54 4f 52 59 0d 00  a3
14a1 : 01 09 08 12 1c c6 4c 4f  33
14a9 : 50 50 59 20 4e 4f 54 20  6d
14b1 : 41 56 41 49 4c 41 42 4c  07
14b9 : 45 20 21 92 1e 00 01 0e  ab
14c1 : 09 b0 ae 20 20 01 0e 0a  28
14c9 : ad bd 20 20 01 0e 0b 20  4e
14d1 : 20 20 20 01 0e 0c 20 20  2c
14d9 : 20 20 00 01 0e 09 b0 c0  97
14e1 : ae 20 01 0e 0a ad c0 bd  2e
14e9 : 20 01 0e 0b 20 20 20 20  33
14f1 : 01 0e 0c 20 20 20 00     84
14f9 : 01 0e 09 b0 c0 c0 ae 01  28
1501 : 0e 0a ad c0 c0 bd 01 0e  b2
1509 : 0b 20 20 20 20 01 0e 0c  8b
1511 : 20 20 20 20 00 01 0e 09  a0
1519 : b0 ae 20 20 01 0e 0a dd  91
1521 : dd 20 20 01 0e 0b ad bd  a2
1529 : 20 20 01 0e 0c 20 20 20  de
1531 : 20 00 01 0e 09 b0 c0 ae  ca
1539 : 20 01 0e 0a dd a0 dd 20  39
1541 : 01 0e 0b ad c0 bd 20 01  3e
1549 : 0e 0c 20 20 20 20 00 01  6f
1551 : 0e 09 b0 c0 c0 ae 01 0e  ca
1559 : 0a dd 20 20 dd 01 0e 0b  92
1561 : ad c0 c0 bd 01 0e 0c 20  47
1569 : 20 20 20 00 01 0e 09 b0  a8
1571 : ae 20 20 01 0e 0a dd dd  bc
1579 : 20 20 01 0e 0b dd dd 20  03
1581 : 20 01 0e 0c ad bd 20 20  b0
1589 : 00 01 0e 09 b0 c0 ae 20  bb
1591 : 01 0e 0a dd 20 dd 20 01  4b
1599 : 0e 0b dd 20 dd 20 01 0e  a7
15a1 : 0c ad c0 bd 20 00 01 0e  8e
15a9 : 09 b0 c0 c0 ae 01 0e 0a  92
15b1 : dd 20 20 dd 01 0e 0b dd  cb
15b9 : 20 20 dd 01 0e 0c ad c0  fa
15c1 : c0 bd 00 01 0e 0d 48 4f  89
15c9 : 52 49 5a 4f 4e 54 41 4c  65
15d1 : 20 00 01 0e 0d 56 45 52  31
15d9 : 54 49 43 41 4c 20 20 20  51
15e1 : 00 01 0e 0d 4c 45 54 54  70
15e9 : 45 52 2d 48 45 41 44 00  1b
15f1 : 01 05 0e 12 cd 41 52 47  f9
15f9 : 49 4e 92 00 01 02 06 cc  e0
1601 : 4f 41 44 20 50 49 43 54  0b
1609 : 55 52 45 20 2e 01 02 07  de
1611 : 28 01 13 07 29 01 03 07  14
1619 : 00 01 0d 0b c9 4e 53 45  25
1621 : 52 54 20 c4 49 53 4b 20  db
1629 : 21 11 0d 00 01 07 09 c3  0a
1631 : 4f 4c 4f 52 52 41 4d 20  69
1639 : 3f a0 28 d9 29 45 53 20  58
1641 : 2f 20 28 ce 29 4f 20 3f  70
1649 : 20 00 01 08 08 12 1c d0  ce
1651 : 52 49 4e 54 45 52 20 4e  6a
1659 : 4f 54 20 41 56 41 49 4c  30
1661 : 41 42 4c 45 20 21 92 1e  11
1669 : 00 bd 37 12 4a 05 a9 85  d6
```

Listing 1. (Schluß)

```
Name : Hardcopy SG 10       c000 c134
--------------------------------------
c000 : 78 a9 0d a0 c0 8d 14 03  73
c008 : 8c 15 03 58 60 a5 c5 c9  c9
c010 : 04 d0 07 a9 00 85 c5 20  f7
c018 : 1d c0 4c 31 ea a2 ff b5  fe
c020 : 00 9d 32 c1 ca d0 f8 a9  1e
c028 : 01 85 cc a9 7e a2 04 a0  a2
c030 : 05 20 ba ff a9 00 20 bd  8a
c038 : ff 20 c0 ff a2 7e 20 c9  aa
c040 : ff a9 1b 20 d2 ff a9 33  19
c048 : 20 d2 ff a9 18 20 d2 ff  d4
c050 : a9 00 ac 88 02 85 15 84  df
c058 : 16 a2 19 20 e1 ff f0 3f  6a
c060 : a0 27 b1 15 c9 20 d0 0b  9a
c068 : 88 10 f7 a9 0d 20 d2 ff  48
c070 : 4c 91 c0 20 09 c1 a0 0b  f0
c078 : b9 1e c1 20 d2 ff 88 d0  a6
c080 : f7 a0 00 a9 00 b1 15 85  ea
c088 : d6 20 b7 c0 c8 c0 28 d0  49
c090 : f2 a9 28 18 65 15 85 15  a3
c098 : 90 02 e6 16 ca d0 bc a9  1f
c0a0 : 0d 20 d2 ff a9 7e 20 c3  08
c0a8 : ff 20 cc ff a2 ff bd 32  70
c0b0 : c1 95 00 ca d0 f8 60 48  7c
c0b8 : 8a 48 98 48 a5 d6 85 f8  af
c0c0 : a9 00 85 f9 06 f8 26 f9  bf
c0c8 : 06 f8 26 f9 06 f8 26 f9  c8
c0d0 : a5 f9 18 65 fa 85 f9 a6  36
c0d8 : 01 a9 80 85 9d a9 00 48  36
c0e0 : a0 07 78 a9 01 29 fb 85  ab
c0e8 : 01 b1 f8 25 9d 86 01 58  67
c0f0 : f0 06 68 18 79 2a c1 48  81
c0f8 : 88 10 e7 68 20 d2 ff 46  b4
c100 : 9d 90 da 68 a8 68 aa 68  f2
c108 : 60 a9 00 85 f9 ad 18 d0  fd
c110 : 29 02 d0 05 a9 d0 85 fa  3c
c118 : 2c a9 d8 85 fa 60 00 01  b4
c120 : 40 04 67 1b 0f 4d 1b 30  c8
c128 : 1b 0d 80 40 20 10 08 04  9d
c130 : 02 01 00 37 00 ff 00 ff  9a
```

Listing 2. Die Anpassung der »Hardcopy auf Knopfdruck« an den STAR SG-10 geben Sie bitte mit dem MSE (Seite 159) ein

```
Name : mini mc-hardcopy     c000 c424
--------------------------------------
c000 : 20 fa c1 a9 01 a2 04 a0  b9
c008 : 00 20 ba ff a9 00 20 bd  5d
c010 : ff 20 c0 ff a2 01 20 c9  96
c018 : ff a9 ea a0 c3 20 1e ab  c7
c020 : a9 27 8d 3c 03 a9 02 8d  e8
c028 : 3e 03 a9 00 8d 58 03 a9  4d
c030 : 20 8d 52 03 a9 40 8d 54  87
c038 : 03 a9 00 8d 3d 03 20 a1  71
c040 : c0 ee 3d 03 ad 3d 03 c9  8b
c048 : c8 d0 f3 ee 58 03 ad 54  50
c050 : 03 c9 80 f0 1e a0 78 a9  92
c058 : 00 20 d2 ff 88 d0 fa 18  48
c060 : ad 54 03 69 20 8d 54 03  eb
c068 : ad 52 03 69 08 8d 52 03  69
c070 : 4c 39 c0 a9 f2 a0 c3 20  42
c078 : 1e ab ce 3e 03 10 ab a9  9a
c080 : f7 a0 c3 a6 02 f0 04 a9  98
c088 : fe a0 c3 20 1e ab ad 3c  3a
c090 : 03 38 e9 03 8d 3c 03 10  71
c098 : 8c 20 cc ff a9 01 4c c3  c3
c0a0 : ff ad 3d 03 4a 4a 4a a8  97
c0a8 : b9 0b c4 18 6d 54 03 85  ac
```

Listing 3. Die Hardcopy-Routine für den Oki 20 geben Sie bitte mit dem MSE (Seite 159) ein

```
c0b0 : fc 98 29 03 a8 b9 07 c4  a1
c0b8 : 38 e9 10 85 fb b0 02 c6  74
c0c0 : fc ad 3d 03 29 07 18 65  39
c0c8 : fb 85 fb ad 3c 03 0a 0a  52
c0d0 : 0a 90 02 e6 fc 18 65 fb  9e
c0d8 : 85 fb 90 02 e6 fc a0 00  98
c0e0 : b1 fb 8d 42 03 a0 08 b1  f3
c0e8 : fb 8d 43 03 a0 10 b1 fb  24
c0f0 : 8d 44 03 a9 00 85 fe ad  19
c0f8 : 3d 03 29 f8 85 fd 0a 26  dd
c100 : fe 0a 26 fe 18 65 fd 85  1c
c108 : fd a9 00 65 fe 85 fe a5  ea
c110 : fd 18 6d 3c 03 08 38 e9  21
c118 : 02 85 fd a5 fe e9 00 28  a0
c120 : 69 00 85 fe a5 fd 85 22  6f
c128 : ad 52 03 18 65 fe 85 fe  25
c130 : 18 69 04 85 23 ad 3e 03  4d
c138 : 0a 0a 0a 0a 0a 8d 4c 03  4f
c140 : a9 02 8d 45 03 a9 00 a0  b5
c148 : 05 99 46 03 88 10 fa a9  54
c150 : 00 8d 4f 03 ac 45 03 b9  c0
c158 : 42 03 8d 4e 03 a9 00 8d  e2
c160 : 4d 03 ad 4e 03 0a 2e 4d  38
c168 : 03 0a 2e 4d 03 8d 4e 03  81
c170 : ac 45 03 ae 4d 03 f0 1a  3a
c178 : e0 03 f0 10 b1 fd e0 01  a8
c180 : f0 04 29 0f 10 12 4a 4a  ee
c188 : 4a 4a 10 0c b1 22 29 0f  6c
c190 : 10 06 ac 58 03 b9 55 03  33
c198 : 0a 18 6d 4c 03 a8 ae 45  4e
c1a0 : 03 b9 5d c3 4a 3e 46 03  05
c1a8 : 4a 3e 46 03 b9 5e c3 4a  36
c1b0 : 3e 49 03 4a 3e 49 03 ee  b5
c1b8 : 4f 03 ad 4f 03 c9 04 d0  0e
c1c0 : 9c ce 45 03 30 05 ad 3c  cf
c1c8 : 03 d0 84 ad 48 03 20 f0  09
c1d0 : c1 ad 47 03 20 f0 c1 ad  86
c1d8 : 46 03 20 f0 c1 ad 4b 03  83
```

```
c1e0 : 20 f0 c1 ad 4a 03 20 f0  bd
c1e8 : c1 ad 49 03 20 f0 c1 60  84
c1f0 : 20 d2 ff c9 03 d0 f8 4c  e6
c1f8 : d2 ff a9 00 8d 50 03 18  cc
c200 : 69 07 aa a9 20 9d c4 c3  56
c208 : ca ec 50 03 10 f7 20 fd  fa
c210 : ae 20 9e ad 20 a3 b6 c9  b9
c218 : 09 90 03 4c 48 b2 a8 18  a1
c220 : 6d 50 03 aa ca 88 b1 22  c7
c228 : 9d c4 c3 ca 88 10 f7 ad  b6
c230 : 50 03 18 69 0f 8d 50 03  da
c238 : c9 1f 90 c3 20 fd ae 20  1a
c240 : 9e b7 e0 02 90 03 4c 48  15
c248 : b2 86 02 a0 00 84 fb a9  39
c250 : 40 85 fc a9 00 a2 60 91  81
c258 : fb c8 d0 fb e6 fc ca 10  0c
c260 : f6 a9 40 8d 51 03 a9 20  01
c268 : 8d 52 03 a9 24 8d 53 03  16
c270 : a9 00 20 ad c2 a5 02 8d  54
c278 : 55 03 a9 60 8d 51 03 a9  88
c280 : 28 8d 52 03 a9 2c 8d 53  3d
c288 : 03 a9 0f 20 ad c2 a5 02  b3
c290 : 8d 56 03 a9 80 8d 51 03  fe
c298 : a9 30 8d 52 03 a9 34 8d  70
c2a0 : 53 03 a9 1e 20 ad c2 a5  69
c2a8 : 02 8d 57 03 60 a2 01 86  d3
c2b0 : 02 a2 00 86 fb ae 51 03  55
c2b8 : 86 fc 18 69 bd aa a9 00  c7
c2c0 : 69 c3 a8 a9 0f 20 bd ff  53
c2c8 : a9 01 a2 08 a0 00 20 ba  9b
c2d0 : ff 20 c0 ff a2 01 20 c6  50
c2d8 : ff 20 cf ff 20 cf ff a2  a1
c2e0 : 1f a0 00 20 cf ff 24 90  02
c2e8 : 70 6b 91 fb e0 00 f0 09  d5
c2f0 : c8 d0 f0 e6 fc ca 4c e3  58
c2f8 : c2 c8 c0 40 d0 e5 a2 00  1d
c300 : 86 fb ae 52 03 86 fc a2  17
c308 : 03 a0 00 20 cf ff 24 90  0e
```

```
c310 : 70 43 91 fb e0 00 f0 09  e9
c318 : c8 d0 f0 e6 fc ca 4c 0b  cf
c320 : c3 c8 c0 e8 d0 e5 a2 00  5b
c328 : 86 fb ae 53 03 86 fc a2  5f
c330 : 03 a0 00 20 cf ff 24 90  36
c338 : 70 1b 91 fb e0 00 f0 09  fd
c340 : c8 d0 f0 e6 fc ca 4c 33  47
c348 : c3 c8 c0 e8 d0 e5 20 cf  19
c350 : ff 29 0f 85 02 20 cc ff  ad
c358 : a9 01 4c c3 ff 03 03 00  31
c360 : 00 02 00 01 02 00 01 03  ac
c368 : 03 03 03 00 00 00 00 02  b2
c370 : 00 00 00 01 03 02 01 02  d9
c378 : 01 02 01 02 00 03 03 00  1f
c380 : 00 03 03 00 00 01 02 00  d3
c388 : 00 02 03 00 00 02 01 02  62
c390 : 01 03 03 01 03 02 01 00  38
c398 : 00 00 00 02 00 03 03 00  fd
c3a0 : 00 03 03 01 00 00 00 03  09
c3a8 : 03 00 00 03 03 03 03 03  66
c3b0 : 03 00 00 01 03 02 01 03  1e
c3b8 : 01 00 00 02 00 81 50 49  da
c3c0 : 43 20 3f 20 20 20 20 20  ab
c3c8 : 20 20 20 20 81 50 49 43  4b
c3d0 : 20 3f 20 20 20 20 20 20  60
c3d8 : 20 20 20 81 50 49 43 20  dd
c3e0 : 3f 20 20 20 20 20 20 20  ff
c3e8 : 20 20 0d 1b 19 1b 52 1c  ab
c3f0 : 03 00 03 02 0d 03 00 03  e3
c3f8 : 02 0a 1b 19 03 00 03 02  2a
c400 : 1b 0a 17 1b 19 03 00 00  f3
c408 : 40 80 c0 00 01 02 03 05  ef
c410 : 06 07 08 0a 0b 0c 0d 0f  41
c418 : 10 11 12 14 15 16 17 19  49
c420 : 1a 1b 1c 1e 00 ff 00 ff  93
```

Listing 3. (Schluß)

```
Name : hard $ce              ce00 cf36
--------------------------------------
ce00 : a9 23 20 ff ae 20 9e b7  19
ce08 : 20 c9 ff 20 db ce a2 0d  ea
ce10 : 86 57 a5 5c a4 5d a2 06  02
ce18 : 20 39 ce 20 f5 ce a5 02  fd
ce20 : d0 f0 a5 5e a4 5f a2 07  7b
ce28 : 20 39 ce 20 f5 ce a5 02  0d
ce30 : d0 f0 c6 57 d0 dc 4c 17  68
ce38 : cf 85 5a 84 5b 85 5e 84  55
ce40 : 5f 18 69 40 90 01 c8 c8  d3
ce48 : 85 5c 84 5d 86 56 a2 4c  06
ce50 : a9 40 a0 01 20 c3 ce a6  0a
ce58 : 57 ca a9 28 85 58 a9 80  46
ce60 : 85 59 a9 00 85 55 78 a5  2c
ce68 : 01 48 a9 30 85 01 a4 56  9d
ce70 : 38 8a f0 06 b1 5c 25 59  2f
ce78 : d0 01 18 66 55 88 88 10  78
ce80 : ef a4 56 38 b1 5a 25 59  93
ce88 : d0 01 18 66 55 88 88 10  88
ce90 : f2 68 85 01 58 a5 55 20  80
ce98 : d2 ff 46 59 90 c4 a5 5a  a1
cea0 : 69 07 85 5a 90 02 e6 5b  a5
cea8 : 18 a5 5c 69 08 85 5c 90  16
ceb0 : 02 e6 5d c6 58 d0 a7 a5  4b
ceb8 : 02 d0 01 60 b9 2f ce a2  04
cec0 : 4a a0 0d 48 a9 1b 20 d2  40
cec8 : ff 8a 20 d2 ff 68 20 d2  d8
ced0 : ff 98 20 d2 ff b0 01 60  c8
ced8 : 4c 04 e1 a9 00 a2 20 a0  ab
cee0 : a0 85 5c 86 5d 8d 32 cf  d5
cee8 : 8c 33 cf a9 00 85 02 a9  be
cef0 : 01 8d 2f cf 60 a5 02 48  49
cef8 : ad 2f cf 85 02 68 8d 2f  d9
cf00 : cf a0 00 b9 5a 00 48 b9  91
cf08 : 30 cf 99 5a 00 68 99 30  dc
cf10 : cf c8 c0 06 d0 ed 60 a0  73
cf18 : 00 b9 27 cf 20 d2 ff c8  e3
cf20 : c0 05 d0 f5 4c cc ff 1b  b7
cf28 : 41 04 1b 0a 00 01 17 00  d8
cf30 : 00 00 00 00 00 00 00 ff  30
```

Listing 4. Die Giga-CAD-Hardcopy im Bereich ab $CE00 geben Sie bitte mit dem MSE (Seite 159) ein

```
Name : hard $90              9000 9136
--------------------------------------
9000 : a9 23 20 ff ae 20 9e b7  19
9008 : 20 c9 ff 20 db 90 a2 0d  f8
9010 : 86 57 a5 5c a4 5d a2 06  02
9018 : 20 39 90 20 f5 90 a5 02  7b
9020 : d0 f0 a5 5e a4 5f a2 07  7b
9028 : 20 39 90 20 f5 90 a5 02  8b
9030 : d0 f0 c6 57 d0 dc 4c 17  68
9038 : 91 85 5a 84 5b 85 5e 84  17
9040 : 5f 18 69 40 90 01 c8 c8  d3
9048 : 85 5c 84 5d 86 56 a2 4c  06
9050 : a9 40 a0 01 20 c3 90 a6  11
9058 : 57 ca a9 28 85 58 a9 80  46
9060 : 85 59 a9 00 85 55 78 a5  2c
9068 : 01 48 a9 30 85 01 a4 56  9d
9070 : 38 8a f0 06 b1 5c 25 59  2f
9078 : d0 01 18 66 55 88 88 10  78
9080 : ef a4 56 38 b1 5a 25 59  93
9088 : d0 01 18 66 55 88 88 10  88
9090 : f2 68 85 01 58 a5 55 20  80
9098 : d2 ff 46 59 90 c4 a5 5a  a1
90a0 : 69 07 85 5a 90 02 e6 5b  a5
90a8 : 18 a5 5c 69 08 85 5c 90  16
90b0 : 02 e6 5d c6 58 d0 a7 a5  4b
90b8 : 02 d0 01 60 b9 2f 90 a2  0b
90c0 : 4a a0 0d 48 a9 1b 20 d2  40
90c8 : ff 8a 20 d2 ff 68 20 d2  d8
90d0 : ff 98 20 d2 ff b0 01 60  c8
90d8 : 4c 04 e1 a9 00 a2 20 a0  ab
90e0 : a0 85 5c 86 5d 8d 32 91  59
90e8 : 8c 33 91 a9 00 85 02 a9  2f
90f0 : 01 8d 2f 91 60 a5 02 48  82
90f8 : ad 2f 91 85 02 68 8d 2f  4a
9100 : 91 a0 00 b9 5a 00 48 b9  53
9108 : 30 91 99 5a 00 68 99 30  bd
9110 : 91 c8 c0 06 d0 ed 60 a0  35
9118 : 00 b9 27 91 20 d2 ff c8  1b
9120 : c0 05 d0 f5 4c cc ff 1b  b7
9128 : 41 04 1b 0a 00 01 17 00  d8
9130 : 00 00 00 00 00 00 00 ff  30
```

Listing 5. Die Giga-CAD-Hardcopy-Routine für den Bereich $E000. Bitte mit dem MSE (Seite 159) eingeben.

```
Name : mp softcolor          c000 c270
--------------------------------------
c000 : 20 fd ae 20 9e b7 86 02  94
c008 : f0 05 20 c9 ff d0 00 20  83
c010 : fd ae 20 9e b7 86 15 a9  98
c018 : 00 85 14 20 fd ae 20 9e  f7
c020 : b7 86 57 20 fd ae 20 9e  07
c028 : b7 86 58 20 fd ae 20 9e  4f
c030 : b7 86 59 20 fd ae 20 9e  98
c038 : b7 86 5a a9 ff 8d 03 dd  32
c040 : ad 02 dd 09 04 8d 02 dd  f7
c048 : a0 02 b9 7c c1 20 59 c1  ed
c050 : 88 10 f7 a9 00 85 62 a9  1c
c058 : 00 85 63 20 e4 ff f0 03  10
c060 : 4c 45 c1 a0 03 b9 78 c1  37
c068 : 20 59 c1 88 10 f7 a9 00  1e
c070 : 85 61 a5 14 a4 15 85 22  df
c078 : 84 23 a4 63 a2 00 78 a9  83
c080 : 35 85 01 b1 22 95 64 c8  e0
c088 : e8 e0 04 d0 f6 a9 37 85  a0
c090 : 01 58 a9 03 85 60 a2 03  74
c098 : b5 64 a4 60 f0 06 4a 4a  b1
c0a0 : 88 4c 9c c0 18 29 03 95  8f
c0a8 : 5b ca 10 ec a0 00 a2 03  a5
c0b0 : a9 00 85 5f a9 03 85 68  40
c0b8 : b5 5b c5 57 d0 06 b9 82  a0
c0c0 : c1 18 90 21 c5 58 d0 06  44
c0c8 : b9 8e c1 18 90 17 c5 59  c7
c0d0 : d0 06 b9 9a c1 18 90 0d  9e
c0d8 : c5 5a d0 06 b9 a6 c1 18  c7
c0e0 : 90 03 b9 b2 c1 25 68 05  a7
c0e8 : 5f 85 5f 18 26 68 18 26  37
c0f0 : 68 18 ca 10 c3 20 59 c1  3f
c0f8 : c8 c0 0c d0 b1 a6 60 ca  a5
c100 : 86 60 10 92 18 a5 22 69  17
c108 : 08 85 22 a5 23 69 00 85  99
c110 : 23 a6 61 e8 86 61 e0 28  43
c118 : f0 03 4c 7a c0 a6 63 e8  8d
c120 : e8 e8 e8 86 63 e0 08 f0  c6
c128 : 03 4c 5b c0 18 a5 14 69  12
c130 : 40 85 14 a5 15 69 01 85  98
```

Listing 6. Der Maschinenspracheteil für den Farbdruck mit Epson-Druckern. Bitte mit dem MSE (Seite 159) eingeben.

```
c138 : 15 a6 62 e8 86 62 e0 19  87
c140 : f0 03 4c 57 c0 a0 02 b9  3c
c148 : 7f c1 20 59 c1 88 10 f7  6c
c150 : 20 cc ff a5 02 20 c3 ff  bb
c158 : 60 a6 02 f0 03 4c d2 ff  88
c160 : 8d 01 dd a9 10 2c 0d dd  6d
c168 : f0 fb ad 00 dd 09 04 8d  13
c170 : 00 dd 29 fb 8d 00 dd 60  3a
c178 : 07 80 5a 1b 18 33 1b 40  c1
c180 : 1b 0d aa 55 aa 55 aa 55  22
c188 : aa 55 aa 55 aa 55 aa 00  32
c190 : 55 00 aa 00 55 00 aa 00  90
c198 : 55 00 00 aa 00 00 55 00  98
c1a0 : 00 aa 00 00 55 00 00 00  4b
c1a8 : aa 00 00 00 00 00 55 00  a8
c1b0 : 00 00 00 00 00 00 00 00  b1
c1b8 : 00 00 00 00 00 00 00 00  b9
c1c0 : 00 00 00 06 07 00 ad 11  cb
c1c8 : d0 09 20 8d 11 d0 ad 18  55
c1d0 ; d0 09 08 8d 18 d0 ad 16  c4
c1d8 : d0 09 10 8d 16 d0 ea ea  4c
c1e0 : a9 00 8d 20 d0 ad c2 c1  fa
c1e8 : 8d 21 d0 ad c3 c1 0a 0a  76
c1f0 : 0a 0a 0d c4 c1 a2 00 86  19
c1f8 : a4 86 a6 a2 04 86 a5 ea  be
c200 : ea ea ea ea ea a2 d8 86  ab
c208 : a7 ae 12 d0 e0 58 d0 f9  ad
c210 : ae 11 d0 30 f4 a0 00 a2  1b
c218 : 08 91 a4 48 ad c5 c1 91  4e
c220 : a6 68 c8 d0 f4 e6 a5 e6  31
c228 : a7 e4 a5 d0 ec 60 ad 16  79
c230 : d0 29 ef 8d 16 d0 ad 11  03
c238 : d0 29 df 8d 11 d0 ad 18  c5
c240 : d0 29 f7 8d 18 d0 a9 0c  1b
c248 : 8d 20 d0 8d 21 d0 a9 93  31
c250 : 20 d2 ff 60 a2 08 a0 01  d4
c258 : 20 ba ff ad 70 c2 a2 71  15
c260 : a0 c2 20 bd ff a9 00 a2  b4
c268 : 00 a0 20 20 d5 ff 60 ea  79
```

Listing 6. (Schluß)

```
10 REM ******************************     <Ø6Ø>
20 REM *                            *     <Ø69>
30 REM *      SOFTCOLOR MIT RX/8Ø   *     <Ø35>
40 REM *                            *     <Ø89>
50 REM *        JUERGEN   CURDT     *     <177>
6Ø REM *        KESSEMEIERWEG 5     *     <Ø8Ø>
7Ø REM *        493Ø  DETMOLD 1     *     <252>
8Ø REM *                            *     <129>
9Ø REM ******************************     <14Ø>
1ØØ POKE 53281,12:POKE 5328Ø,12           <169>
11Ø IF I=Ø THEN I=1:LOAD"MP SOFTCOLOR",8,1 <Ø81>
12Ø FA=49632:GR=496Ø6:TE=4971Ø:LA=49748   <235>
13Ø PRINT"{CLR,BLUE}":Y=12:GOSUB 77Ø      <185>
14Ø INPUT"WELCHE GRAFIK LADEN ";A$        <155>
15Ø IF LEN(A$)>12 THEN A$=LEFT$(A$,12)    <Ø39>
16Ø POKE LA+28,LEN(A$)                    <Ø53>
17Ø FOR I=1 TO LEN(A$):POKE LA+28+I,ASC(MI
    D$(A$,I,1)):NEXT                      <134>
18Ø PRINT"{CLR}":Y=8:X=2:GOSUB 77Ø        <216>
19Ø PRINT"{WHITE}↑{2SPACE,BLUE}UMSCHALTEN
    GRAFIK <> MENUE":GOSUB 76Ø            <Ø88>
2ØØ PRINT"{WHITE}*{2SPACE,BLUE}ANDERE{2SPA
    CE}FLAECHE{2SPACE}ANWAEHLEN":X=1:GOSUB
     76Ø                                  <174>
21Ø PRINT"{WHITE}+/-{SPACE,BLUE}BILDSCHIRM
    FARBE UMSCHALTEN{GREY 2}"             <162>
22Ø SYS LA:OPEN 1,8,15:INPUT#1,I:CLOSE 1  <12Ø>
23Ø IF I<>Ø THEN 13Ø                      <ØØ5>
24Ø X=5:Y=18:GOSUB 77Ø:PRINT"{BLACK}BITTE
    TASTE DRUECKEN!{GREY 2}"              <236>
25Ø POKE 198,Ø:WAIT 198,1                 <222>
26Ø FOR I=Ø TO 3:F(I)=496Ø2+I:POKE F(I),7+
    I:NEXT                                <152>
27Ø FOR I=Ø TO 3:FD(I)=5:FM(I)=7+I:NEXT   <222>
28Ø F1=FD(Ø):F2=FD(1):F3=FD(2):F4=FD(3)   <Ø67>
29Ø F=Ø:SYS GR                            <184>
3ØØ I=Ø                                   <235>
31Ø I=I+1:GET A$:IF A$=""AND I<5Ø THEN 31Ø <141>
32Ø IF A$<>"*"THEN 35Ø                    <173>
33Ø FOR I=Ø TO 3:POKE F(I),FM(I):NEXT     <Ø55>
34Ø F=F+1:IF F>3 THEN F=Ø                 <162>
35Ø IF A$="+"THEN FM(F)=(FM(F)+1)AND 15:PO
    KE F(F),FM(F)                         <2Ø2>
36Ø IF A$="-"THEN FM(F)=(FM(F)-1)AND 15:PO
    KE F(F),FM(F)                         <Ø21>
37Ø IF PEEK(F(F))=Ø THEN POKE F(F),FM(F):G
    OTO 39Ø                               <Ø99>
38Ø IF PEEK(F(F))<>Ø THEN POKE F(F),Ø     <2Ø2>
39Ø SYS FA                                <Ø72>
4ØØ IF A$<>"↑"THEN 3ØØ                    <225>
41Ø SYS TE                                <Ø12>
42Ø PRINT TAB(81)"{BLACK}DIE IN DER GRAFIK
     BLINKENDEN FLAECHEN"                 <175>
43Ø PRINT TAB(44)"SOLLEN AUSGEDRUCKT WERDE
    N MIT:"                               <ØØ6>
44Ø X=8:Y=4                               <Ø93>
45Ø IF F1=5 THEN GOSUB 76Ø:PRINT"{WHITE}4{
    BLUE,2SPACE}GROESSTER FARBDICHTE"     <174>
46Ø IF F2=5 THEN GOSUB 76Ø:PRINT"{WHITE}3{
    BLUE,2SPACE}GROSSER DICHTE"           <12Ø>
47Ø IF F3=5 THEN GOSUB 76Ø:PRINT"{WHITE}2{
    BLUE,2SPACE}GERINGER DICHTE"          <17Ø>
48Ø IF F4=5 THEN GOSUB 76Ø:PRINT"{WHITE}1{
    BLUE,2SPACE}KLEINSTER FARBDICHTE"     <121>
49Ø X=1:GOSUB 76Ø:PRINT"{BLACK}BITTE ZIFFE
    R FUER FARBDICHTE WAEHLEN             <159>
5ØØ PRINT"{DOWN,6SPACE}ODER{SPACE,BLUE}UMS
    CHALTEN ZUR GRAFIK{SPACE,WHITE}↑      <219>
51Ø GET A$:IF A$="↑"THEN A$="":GOTO 29Ø   <169>
52Ø IF A$<"1"OR A$>"4"THEN 51Ø            <152>
53Ø FD(4-VAL(A$))=F                       <Ø33>
54Ø F1=FD(Ø):F2=FD(1):F3=FD(2):F4=FD(3)   <Ø73>
55Ø GOSUB 78Ø:PRINT"{WHITE}P{BLUE,3SPACE}A
    USDRUCK STARTEN                       <199>
56Ø GET A$:IF A$="↑"THEN SYS GR:A$="":GOTO
     34Ø                                  <114>
57Ø IF A$="N"THEN 13Ø                     <142>
58Ø IF A$<>"P"THEN 56Ø                    <249>
59Ø PRINT"{CLR}":X=3:Y=5:GOSUB 77Ø        <Ø84>
6ØØ PRINT"{BLACK}BITTE VOR DEM START CHECK
    EN:{BLUE}"                            <Ø86>
61Ø PRINT SPC(126)"* IST DER DRUCKER ON LI
    NE?                                   <157>
62Ø PRINT SPC(46)"* IST DIE STARTPOSITION
    AUF DEM"                              <Ø21>
63Ø PRINT TAB(8)"PAPIER MARKIERT? - IST DA
    S"                                    <25Ø>
64Ø PRINT TAB(8)"PAPIER ZUR STARTMARKIERUN
    G"                                    <25Ø>
65Ø PRINT TAB(8)"ZURUECKGEDREHT?"         <233>
66Ø PRINT TAB(46)"* IST DIE DRUCKFARBE GEW
    ECHSELT?                              <238>
67Ø PRINT TAB(123)"{BLACK}ALLES FERTIG? BI
    TTE TASTE DRUECKEN!"                  <199>
68Ø GET A$:IF A$=""THEN 68Ø               <2Ø7>
69Ø SYS GR                                <143>
7ØØ OPEN 4,4:SYS 49152,4,32,F1,F2,F3,F4:CL
    OSE 4                                 <126>
71Ø SYS TE:GOSUB 78Ø                      <Ø23>
72Ø GET A$:IF A$="N"THEN 13Ø              <Ø29>
73Ø IF A$<>"↑"THEN 72Ø                    <1Ø9>
74Ø A$="":GOTO 27Ø                        <1Ø8>
75Ø REM UP'E CURSOR POSITIONIEREN PP      <219>
76Ø Y=Y+3                                 <172>
77Ø POKE 211,X:POKE 214,Y:SYS 58732:RETURN <Ø75>
78Ø PRINT"{CLR,BLACK}":X=1Ø:Y=8:GOSUB 77Ø <254>
79Ø PRINT"BITTE WAEHLEN:":X=6:GOSUB 76Ø   <15Ø>
8ØØ PRINT"{WHITE}↑{BLUE,3SPACE}UMSCHALTEN
    ZUR GRAFIK":GOSUB 76Ø                 <164>
81Ø PRINT"{WHITE}N{BLUE,3SPACE}NEUE GRAFIK
     LADEN"                               <Ø29>
82Ø GOSUB 76Ø:RETURN                      <1Ø3>
83Ø **************  END E  *************  <222>
```


Listing 7. Das Basic-Programm für den Epson-Farbdruck geben Sie bitte mit dem Checksummer (Seite 159) ein

Auf der Suche nach der Grafik

Egal, wo Ihre HiRes-Grafik liegt, »Hardmaker« findet sie. Multi-Color-Grafiken können auf Wunsch in Graustufen umgerechnet und auf dem MPS 801 oder Epson-Drucker ausgedruckt werden.

Der »Hardmaker« erlaubt es, HiRes-Grafiken aus fast allen Programmen aufs Papier zu bringen. Dazu stehen umfangreiche Routinen zur Verfügung, die dem Benutzer fast alle Arbeiten abnehmen. Nur das Papier müssen Sie noch von Hand einspannen.

Eingabehinweise:

Geben Sie den »Hardmaker« (Listing 1) für den MPS 801/803 bitte mit dem MSE ein, und speichern Sie ihn. Der Programmstart erfolgt durch RUN.

Danach sehen Sie auf dem Bildschirm (meistens) ein wüstes Durcheinander von Punkten. Das ist der Bereich von $2000 bis $4000 als Multicolor-Grafik dargestellt. Das könnte zum Beispiel Teil eines Computerspiels sein, das vorher im Computer war. Wenn Sie nun ein Programm auf Bilder untersuchen wollen, müßten Sie im »Diskworkmodus« dieses Programm laden, die Grafik finden, eventuell speichern und ausdrucken. Dazu stehen Ihnen folgende Funktionen zur Verfügung.

Speicherbereiche:

Computergrafiken können nur an bestimmten Stellen im Speicher stehen, um vom VIC ausgelesen werden zu können. Ein solcher Bereich ist der von $2000 bis $3FFF. Diesen Bereich sehen Sie grundsätzlich auf dem Bildschirm; er wird vom Programm als Grafik-RAM benutzt. Wollen Sie den Inhalt eines anderen Bereiches sehen, muß er nach $2000 transportiert werden. Dazu dienen die Tasten ⟨1⟩ bis ⟨6⟩ und ⟨←⟩:

⟨1⟩ :	$4000 bis $5FFF
⟨2⟩ :	$6000 bis $7FFF
⟨3⟩ :	$8000 bis $9FFF
⟨4⟩ :	$A000 bis $BFFF (RAM unterm Basic)
⟨5⟩ :	$C000 bis $DFFF ($D000 bis $DFFF; RAM unter I/O)
⟨6⟩ :	$E000 bis $FFFF (RAM unterm Kernel)
⟨←⟩:	$0000 bis $1FFF, Dieser Bereich ist nur der Vollständigkeit halber per Taste zerreichbar. Benutzen können Sie ihn nicht, da dort Zeropage, Stack, Video-RAM und der Hardmaker selbst liegen!

Wenn Sie auf eine dieser Tasten ohne ⟨SHIFT⟩, ⟨CTRL⟩ oder ⟨CBM⟩ drücken, wird der entsprechende Speicherbereich nach $2000 transportiert und ist damit auf dem Bildschirm sichtbar. Drücken Sie jedoch ⟨SHIFT⟩ und eine dieser Tasten, wird der entsprechende Bereich mit dem ab $2000 ODER-verknüpft. So können zwei Bilder zusammengemischt werden. Das Ergebnis liegt wieder ab $2000 im Speicher.

Folgende Kombinationen bewirken also:

⟨←⟩ bis ⟨6⟩	≙ jeweiligen Bereich nach $2000 kopieren
⟨SHIFT⟩	≙ ODER-Verknüpfen
⟨CBM⟩	≙ EX-OR-Verknüpfen
⟨SHIFT CBM⟩	≙ UND-Verknüpfen
⟨CTRL⟩	≙ Bereich mit dem ab $2000 vertauschen
⟨CTRL SHIFT⟩ oder ⟨CTRL CBM⟩	≙ $2000 bis $3FFF in entsprechenden Bereich kopieren

Wenn Sie eine Kombination mit ⟨CTRL⟩ drücken, wird der entsprechende Speicherplatz verändert! Auf diese Weise können Sie zum Beispiel den Inhalt von $2000 bis $3FFF zwischenspeichern, wenn Sie ihn danach weiterbearbeiten wollen (Zum Beispiel bei schwierigen Korrekturen).

Bilder »schneiden«

Manchmal kommt es vor, daß nicht der gesamte Inhalt des Bildschirms zu einer Grafik gehört und man den Rest »wegschneiden« möchte. Zum Beispiel wollen Sie am rechten Rand etwas entfernen. Dazu drücken Sie die Taste ⟨R⟩. Der Rahmen wechselt seine Farbe und an der rechten Seite erscheint eine flackernde Linie, die Sie mit ⟨CURSOR-links/-rechts⟩ hin- und herbewegen können.

Wenn Sie die Linie richtig positioniert haben, drücken Sie auf ⟨SPACE⟩; der Bereich rechts der Linie wird gelöscht (oder gefüllt, wenn sie ⟨SHIFT SPACE⟩ drücken). Möchten Sie nichts löschen, dann drücken Sie ⟨Q⟩, und die flackernde Linie ist verschwunden.

⟨R⟩ :	rechts	Randfarbe: orange
⟨L⟩ :	links	Randfarbe: blau
⟨O⟩ :	oben	Randfarbe: hellrot
⟨U⟩ :	unten	Randfarbe: grün

Im Schneidemodus stehen folgende Funktionen zur Verfügung:

— Cursor nach oben/unten (nur ⟨O⟩, ⟨U⟩)
— Cursor nach oben links/rechts (nur ⟨L⟩, ⟨R⟩)
— ⟨SPACE⟩, ⟨SHIFT SPACE⟩
— ⟨Q⟩ (wie quit)

Bilder verschieben:

Wenn die Grafik nicht genau oben links beginnt, muß sie verschoben werden. Eine Möglichkeit dazu ist das Scrollen:

Mit CBM-Taste + Cursor-Taste wird die Grafik um 1 Byte nach links oder rechts verschoben.

Die andere Möglichkeit sind die Tasten ⟨A⟩ und ⟨E⟩. Positionieren Sie den Cursor irgendwo mitten auf dem Bildschirm und drücken Sie ⟨A⟩: Die Grafik wird so verschoben, daß die Cursorposition nun den Anfang der Grafik bildet. Analog funktioniert hier die Taste ⟨E⟩: Die Cursorposition bildet jetzt das Ende der Grafik. Diese Funktionen ermöglichen ein bequemes Positionieren einer Grafik, die irgendwo im Speicher liegt.

Farbe

Mit den Funktionstasten kann die Farbe der Grafik geändert werden, wenn die Ausgangsbelegung (die sich für Schwarzweiß-Fernseher übrigens gut eignet) nicht gefällt:

⟨F1⟩/⟨F2⟩ ≙	Farbe 1 HiRes/Multi
⟨F3⟩/⟨F4⟩ ≙	Farbe 2 HiRes/Multi
⟨F5⟩/⟨F6⟩ ≙	Farbe 3 Multi
⟨F7⟩/⟨F8⟩ ≙	Farbe 4 Multi

Die Funktionstaste blättert die Farben vorwärts, geSHIFTet blättert sie die Farben zurück. Für Multi und HiRes sind getrennte Farb-Speicher vorhanden.

Folgende Tasten bewegen den Cursor:

Cursortasten	gewohnte Cursor-Bewegung
⟨RETURN⟩	Bewegt ihn in die erste Spalte der nächsten Zeile

⟨SHIFT CLEAR/HOME⟩	löscht die Grafik
⟨CLEAR/HOME⟩	Cursor links oben
⟨£⟩	(Gegenteil von HOME: Cursor in letzte Spalte letzte Zeile)
⟨SPACE⟩	Cursor rechts + Cursorfeld löschen
⟨SHIFT SPACE⟩	Cursor rechts + Cursorfeld füllen mit Farbe 3
⟨CBM SPACE⟩	Cursor rechts + Cursorfeld füllen mit Farbe 1
⟨CTRL SPACE⟩	Cursor rechts + Cursorfeld füllen mit Farbe 2
⟨INST/DEL⟩	genauso wie SPACE, bloß mit Cursor nach links (also kein echtes DEL!)

Sonderfunktionen:

⟨H⟩	schaltet HiRes-Modus ein; genauso wie Sie die Grafik nun sehen, wird sie von einem Matrixdrucker ausgegeben werden.
⟨M⟩	schaltet Multicolor an.
⟨T⟩	ist eine sehr praktische Sache, wenn man einen der unter »Hardcopy« erwähnten Matixdrucker besitzt. Diese geben die Multicolor-Grafiken nämlich so aus, daß die Farben 1 und 2 als charakteristische Linien erscheinen. T verwandelt diese Farben nun in Graustufen, die der Drucker ausgeben kann (siehe auch Textkasten). T funktioniert nur, wenn Multicolor eingeschaltet ist, und schaltet dann auf HiRes um!
⟨I⟩	invertiert die Grafik
⟨S⟩	spiegelt die Grafik an der Vertikalen, und vertauscht die Farben 1 und 2, die ja ebenfalls gespiegelt werden. Spiegeln an der Horizontalen ist nicht nötig, da man ja die Hardcopy einfach umdrehen kann!
⟨X⟩	kehrt ins Basic zurück.

Funktionsweise des T-Befehls beim Hardmaker

Im Multicolor-Modus stellen immer je 2 Bit eines Bytes die Information für einen Bildpunkt zur Verfügung, so daß die Auflösung gegenüber dem HiRes-Modus halbiert wird.

Farbige Grafiken in Graustufen umsetzen

Dafür hat man vier Farben statt zwei zur Auswahl, die durch die Bitkombinationen 00, 01, 10 und 11 repräsentiert werden. Ein normaler Matrixdrucker hat hingegen keinen Multicolor-Modus und stellt die Bitmuster genau so dar, wie sie im Grafik-ROM stehen. Das führt bei Flächen, die aus den Farben 1 und 2 beziehungsweise den Bitmustern 01 oder 10 bestehen, zu den bekannten Streifenmustern (Bild 1). Um diese in Graustufen (Bild 2) umzuwandeln, geht der Computer folgendermaßen vor: Die vier Zwei-Bit-Kombinationen eines jeden der 8000 Byte des Grafik-RAMs werden nach den Bitkombinationen 01 und 10 untersucht.

1. Fall: Zwei-Bit-Kombination = 00 oder 11: nichts wird verändert.

2. Fall: Zwei-Bit-Kombination = 01: In diesem Fall werden die zwei Bit invertiert, also durch 10 ersetzt, aber nur dann, wenn die zwei Bit Bestandteil des 1., 3., 5., etc. Bytes des Grafik-RAMs sind. Dadurch wird das Bitmuster in jeder zweiten Grafikzeile um ein Bit nach links versetzt. Wenn die ersten 3 Byte also

```
01 01 00 01                              10 10 00 10
01 01 11 01  lauten, so wird daraus      01 01 11 01
01 01 01 01                              10 10 10 10
```

und der Drucker gibt ein gleichmäßig graues Feld aus.

3. Fall: Bit-Kombination = 10

Hierbei würde aus den ersten 3 Byte

```
10 10 00 10                              01 10 00 10
10 10 11 10  folgendes werden:           10 01 11 01
10 10 10 10                              01 10 01 10
```

Also ein etwas gröberes Graufeld. Dieser Effekt wird dadurch erreicht, daß eine Zwei-Bit-Kombination nur dann invertiert wird, wenn sie entweder im 1., 3., 5., ... Byte die 1. oder 3. Zwei-Bit-Kombination oder im 2., 4., 6., ... Byte die 2. oder 4. Zwei-Bit-Kombination ist.

Diskworkmodus:

Wird mit ⟨D⟩ aktiviert. Die Grafik wird dann ausgeblendet und Sie haben den normalen Kernel-Editor vor sich. Folgende Befehle wurden hier implementiert.

$	Directory
@	Kommandokanal des Laufwerks abfragen
@i	initialisieren (analoges gilt für SCRATCH, RENAME, FORMAT, etc.)
n	gibt die aktuelle Geräteadresse aus (Voreingestellt 8)
n9	schaltet auf Geräteadresse 9 um
?	arbeitet wie der normale Basic-PRINT-Befehl, kann aber hier für Berechnung genutzt werden



Bild 1. Eine Hardcopy ohne Umrechnung...

Q	schaltet den Grafikmodus wieder an
X	Rückkehr zum Basic
←	Mit ← kann aus dem Hardmaker heraus eine Grafik gespeichert werden; zum Beispiel »←Bild 1 ⟨RETURN⟩« speichert den Bereich $2000 bis $3FFF (genauer bis $3F40) auf Diskette unter dem Namen »Bild 1« ab. Das erzeugte Programmfile ist 32 Blöcke lang.
£	Um ein Programm nach Grafiken zu durchsuchen, muß man es irgendwie in den Computer bekommen. Die eine Möglichkeit ist, das Programm zu laden, zu starten, mit Reset auszusteigen und dann den Hardmaster nachzuladen. Dies empfiehlt sich immer dann, wenn das Programm die Grafik erst erzeugt. Außerdem liegt diese dann meist an der richtigen Stelle im Speicher, so daß man nicht mehr zu verschieben braucht. Die andere Möglichkeit ist, erst den Hardmaker zu laden und dann mit dem £-Befehl das betreffende Programm (mit £ Name ⟨RETURN⟩). Dies wird dann in den Bereich ab $2000 geladen, so daß ein Autostart entfällt. Diese Methode kann zu Problemen führen, wenn das betreffende File länger als 178 Blöcke ist, weil dann in die Register der I/O-Bausteine geladen wird. Diese Methode ist dann nötig, wenn ein Programm sich nicht durch Reset stoppen läßt.

Hardcopy

Dieser Programmteil ist zugleich der wichtigste wie auch der problematischste. Denn eine Hardcopy-Routine muß an die meisten Drucker speziell angepaßt werden. Die im »Hardmaker« integrierte Hardcopy-Routine spricht Drucker mit sieben Nadeln an, also den MPS 801, den MPS 803, und Epson-Drucker mit entsprechendem Interface. Für Letztere hilft nur ausprobieren oder eine andere Hardcopy-Routine. Diese kann ab der Adresse $13A0 an den »Hardmaker« angehängt werden. Dazu kommen wir aber später noch.

Aktiviert wird die Hardcopy mit der Taste P. Daraufhin färbt sich der Rahmen schwarz und der C 64 beginnt mit der Hardcopy. Danach wird in den Grafikmodus zurückgesprungen.

»Hardmaker« für Epson-Drucker

Warum soll man sich mit einem Epson-Drucker auf die Grafikfähigkeiten eines MPS 801/803 beschränken? Die grafischen Möglichkeiten eines Epson-kompatiblen Druckers erlauben es außerdem, Multicolor-Grafiken durch Zuordnung von Graustufen auszugeben. Dadurch kann eine noch realistischere Wiedergabe in Graustufen (»T-Befehl«) geboten werden.



Bild 2. ...und eine mit Umrechnung der Multi-Color-Daten.

Für die Ausgabe von HiRes-Grafiken wurde bei »Multiprint« darauf Wert gelegt, daß sowohl Bilder im HiRes-Modus als auch Grafiken im Multicolor-Modus gedruckt werden können. Um eine möglichst große Flexibilität zu erreichen, wurden Hardcopy-Routinen gewählt, die sowohl Drucker über den seriellen IEC-Bus als auch über eine Centronics-Schnittstelle bedienen können.

An der Bedienung hat sich nichts geändert. Der Start des Druckerprogramms erfolgt durch Drücken des Taste ⟨P⟩. Dabei erfolgt der Ausdruck im HiRes-Modus, wenn die Bildschirm-Darstellung ebenfalls HiRes darstellt. Wurde zuvor mit Taste ⟨M⟩ Multicolor angewählt, so wird eine Umsetzung der Bildschirm-Darstellung in Graustufen durchgeführt. Dabei wird der Helligkeitseindruck des momentan angezeigten Bildes zugrundegelegt. Die Parameterwerte für die Hardcopy-Routine ermittelt das Programm automatisch. Jeder der 16 möglichen Farben ist über eine Tabelle der Wert 0 (weiß), 1 (hellgrau), 2 (dunkelgrau) und 3 (schwarz) zugeordnet. Aus diesem Grund empfiehlt es sich, das Bild auf einem monochromen Monitor oder einem Schwarzweiß-Fernseher zu betrachten (Farbmonitor: Farbintensität zurückdrehen).

Hinweise zum Eintippen und zur Drucker-Anpassung

Geben Sie das neue Druckprogramm (Listing 2) bitte mit dem MSE ein und speichern Sie es. Laden Sie dann den »Hardmaker«. Danach laden Sie das neue Druckprogramm mit LOAD »MULTIPRINT«,8,1. Jetzt müssen Sie noch die Zeiger für den Programmbeginn und das neue Programm-Ende korrigieren. Dies erreichen Sie mit folgenden POKE-Befehlen im Direktmodus:

```
POKE 43,1:POKE 44,8:POKE 45,30:POKE 46,23
```

In der abgedruckten Form arbeitet das Programm mit Druckern zusammen, die am seriellen Port des Commodore 64 angeschlossen und über Geräteadresse 4, Sekundäradresse 1 im Linearmodus betrieben werden. Die Geräteadresse ist im Programm in der Speicherstelle 5866 ($16EA) abgelegt. Die Sekundäradresse steht in Speicherstelle 5867 ($16EB).

Wollen Sie dagegen einen Drucker mit Parallelschnittstelle direkt am User-Port betreiben, so geben Sie bitte »POKE 5410,0« ein. Dadurch wird der ebenfalls im Programm enthaltene Centronics-Druckertreiber aktiviert.

Wenn Sie noch ein übriges tun wollen, so sollten Sie die Voreinstellungen für die Multicolor-Farben ändern, damit Sie auch auf einem Farbmonitor eine Darstellung in Grauwerten erhalten. Sie müssen dazu lediglich zwei POKE-Befehle geben:

```
POKE 2088,1:POKE 2100,252
```

Jetzt können Sie die neue Version des Hardmaker speichern. (C. Kurts/M. Wilhelm/kn)

```
programm : hardmaker          0801 1500
----------------------------------------

0801 : 25 08 c0 07 9e 20 32 30   4f
0809 : 38 37 20 20 09 0e 05 57   ad
0811 : 52 49 54 54 45 4e 20 42   73
0819 : 59 20 43 48 52 2e 4b 55   cb
0821 : 52 54 53 00 00 00 a9 0f   37
0829 : 85 fc 8d de 03 a0 00 8c   ba
0831 : dd 03 a9 cb 85 fb a9 20   93
0839 : 8d e0 03 a9 00 8d df 03   1e
0841 : a9 e8 8d 00 03 a9 10 8d   9b
0849 : 01 03 a9 08 8d e2 03 a9   86
0851 : 80 85 9d ad 11 d0 29 bf   6d
0859 : 09 38 8d 11 d0 ad 16 d0   78
0861 : 09 18 8d 16 d0 ad 18 d0   19
0869 : 09 08 8d 18 d0 20 4d 09   32
0871 : a9 13 20 d2 ff a9 0f 8d   ab
0879 : 20 d0 20 14 0b 85 40 8a   7f
0881 : 18 69 20 85 41 a2 02 a0   79
0889 : 07 b1 40 49 ff 91 40 88   41
0891 : 10 f7 ca d0 f2 a5 c6 d0   83
0899 : 09 a6 cb e8 f0 dc a2 0d   a0
08a1 : d0 28 78 20 b4 e5 a2 24   f5
08a9 : dd de 08 d0 14 48 8a 0a   d3
08b1 : a8 b9 04 09 8d c1 08 b9   d3
08b9 : 03 09 8d c0 08 68 4c 00   b1
08c1 : 00 e0 0f f0 05 ca 10 e0   bc
08c9 : 30 ab a4 cb b9 81 eb cd   64
08d1 : e1 03 d0 04 a9 ff d0 eb   9e
08d9 : 8d e1 03 d0 e6 5f 31 31   c2
08e1 : 32 33 34 35 36 85 86 87   19
08e9 : 88 20 14 4d 48 58 93 13   ec
08f1 : 49 53 41 45 c5 5c 0d 11   72
08f9 : 1d 91 9d 4c 52 4f 55 44   4d
0901 : 50 54 82 09 82 09 82 09   ca
0909 : 82 09 82 09 82 09 82 09   5e
0911 : 82 09 6c 0b a3 0b cb 0b   6c
0919 : e3 0b fb 0b fb 0b 1f 0d   91
0921 : 36 0d 1e 0c 5d 0c 4d 0b   68
0929 : 6c 0c 89 0c c1 0a aa 0a   aa
0931 : aa 0a 62 0b 53 0b 47 0b   9b
0939 : 41 0b 2d 0b 37 0b 51 0d   d8
0941 : 49 0d ea 0e f2 0e 8e 10   87
0949 : 74 13 0a 13 a2 00 ad de   ca
0951 : 03 8d 21 d0 ad 16 d0 29   9e
0959 : 10 f0 03 a5 fb 2c a5 fc   08
0961 : 9d 00 04 9d 00 05 9d 00   52
0969 : 06 9d 00 07 48 ad dd 03   8e
0971 : 9d 00 d8 9d 00 d9 9d 00   3d
0979 : da 9d 00 db 68 e8 d0 e0   70
0981 : 60 ac 8d 02 a9 3f 85 3c   fe
0989 : bd ea 09 85 8c 48 38 e9   6e
0991 : 1f 8d e0 03 a9 00 8d df   a0
0999 : 03 68 c0 04 f0 2f 90 0e   68
09a1 : ca 30 1e 85 3c a9 01 8d   ec
09a9 : 20 d0 a9 3f 85 8c b9 f2   0d
09b1 : 09 8d c7 09 a9 c5 8d 0d   ad
09b9 : 0a a9 09 8d 0e 0a 20 fa   34
09c1 : 09 4c 76 08 b1 8b 91 3b   c3
09c9 : 91 3b 4c 0f 0a ca 30 f1   88
09d1 : a9 dd 8d 0d 0a a9 09 8d   9b
09d9 : 0e 0a d0 e2 b1 8b 48 b1   79
09e1 : 3b 91 8b 68 91 3b 4c 0f   17
09e9 : 0a 1f 3f 5f 7f 9f bf df   f2
09f1 : ff 64 11 51 31 65 44 04   e8
09f9 : d4 78 a9 34 85 01 38 a5   87
0a01 : 3c e9 20 85 40 a0 00 84   fd
0a09 : 8b 84 3b 4c 00 00 88 d0   f3
0a11 : fa c6 8c a5 40 c7 3c d0   1b
0a19 : f2 a9 37 85 01 58 60 a9   06
0a21 : ff cf df 03 d0 03 ce e0   82
0a29 : 03 a9 1f 85 60 a2 f8 86   85
0a31 : 5f a2 3f 86 5b 86 59 a2   17
0a39 : 47 86 5a e8 86 58 20 bf   a2
0a41 : a3 a9 00 8d f8 1f 4c 76   11
0a49 : 08 ee df 03 d0 03 ee e0   c3
0a51 : 03 a9 1f 85 60 85 59 a2   7e
0a59 : f8 86 58 e8 86 5f a2 3f   34
0a61 : 86 5b a2 48 86 5a 20 71   e5
0a69 : 0a a9 00 8d 47 3f f0 d6   d9
0a71 : 38 a5 5a e5 5f 85 fd 49   7c
0a79 : ff a8 a5 5b e5 60 aa e8   7f
0a81 : 18 a5 5f 65 fd 85 5f a5   c5
```

Listing 1. »Hardmaker« für MPS 801/803. Bitte mit dem MSE (Seite 159) eingeben.

```
0a89 : 60 e9 00 85 60 18 a5 58  9d
0a91 : 65 fd 85 58 a5 59 e9 00  2e
0a99 : 85 59 b1 5f 91 58 c8 d0  c4
0aa1 : f9 e6 60 e6 59 ca d0 f2  17
0aa9 : 60 20 14 0b a0 27 84 8b  ec
0ab1 : a0 18 84 8c 38 e9 38 48  54
0ab9 : 8a e9 1f aa 68 4c ca 0a  7d
0ac1 : 20 14 0b a0 00 84 8b 84  1e
0ac9 : 8c 18 6d df 03 8d df 03  db
0ad1 : 85 5a 85 5f 8a 6d e0 03  6e
0ad9 : 8d e0 03 85 60 18 69 20  f5
0ae1 : 85 5b a9 00 85 58 a9 20  80
0ae9 : 85 59 ad e0 03 c9 20 90  c2
0af1 : 0b 78 a9 34 85 01 20 71  ed
0af9 : 0a 4c 04 0b a9 40 85 59  f1
0b01 : 20 bf a3 a9 37 85 01 58  73
0b09 : a6 8c a4 8b 18 20 f0 ff  d6
0b11 : 4c 50 0b a5 d1 18 65 d3  18
0b19 : aa a5 d2 e9 03 85 fe 8a  f5
0b21 : 0a 26 fe 0a 26 fe 0a 26  0e
0b29 : fe a6 fe 60 ae 8d 02 e0  67
0b31 : 02 90 19 4c 4a 0a ae 8d  16
0b39 : 02 e0 02 90 0f 4c 20 0a  26
0b41 : a6 d3 e0 27 d0 06 a6 d6  73
0b49 : e0 18 f0 03 20 d2 ff 4c  03
0b51 : 76 08 a6 d6 e0 18 d0 f4  4c
0b59 : a9 91 20 d2 ff a9 0d d0  50
0b61 : eb a0 27 a2 18 18 20 f0  5f
0b69 : ff d0 e4 a9 e8 ac 8d 02  6d
0b71 : f0 02 a9 ca 8d 92 0b ad  1b
0b79 : 16 d0 29 10 f0 03 a9 fb  09
0b81 : 2c a9 fc 8d 8e 0b 8d 98  1b
0b89 : 0b 8d 9c 0b a5 01 29 0f  08
0b91 : aa 9a 8a 09 f0 aa a5 01  49
0b99 : 09 0f 87 00 20 4d 09 4c  35
0ba1 : 76 08 a9 10 ac 8d 02 f0  a9
0ba9 : 02 a9 f0 8d c5 0b ad 16  05
0bb1 : d0 29 10 f0 03 a9 fb 2c  fe
0bb9 : a9 fc 8d c2 0b 8d c7 0b  ee
0bc1 : a5 01 18 69 00 85 01 4c  e3
0bc9 : 9d 0b ad 16 d0 29 10 f0  92
0bd1 : cb ac 8d 02 d0 06 ee dd  4b
0bd9 : 03 4c 9d 0b ce dd 03 4c  4c
0be1 : 9d 0b ad 16 d0 29 10 f0  aa
0be9 : b3 ac 8d 02 d0 06 ee de  4d
0bf1 : 03 4c 9d 0b ce de 03 4c  6c
0bf9 : 9d 0b ac 8d 02 b9 16 0c  57
0c01 : a0 07 91 40 88 10 fb e0  4c
0c09 : 0d f0 05 a9 1d 4c 41 0b  54
0c11 : a9 9d 4c 4d 0b 00 ff 55  a1
0c19 : aa 3c cc 33 c3 20 30 0c  91
0c21 : a9 8b 8d 00 03 a9 e3 8d  1b
0c29 : 01 03 68 68 4c 7b e3 a9  56
0c31 : 00 85 c6 20 44 e5 ad 11  f6
0c39 : d0 29 df 8d 11 d0 a9 06  92
0c41 : 8d 21 d0 ad 16 d0 29 ef  b5
0c49 : 8d 16 d0 a9 0e 8d 20 d0  ba
0c51 : 8d 86 02 ad 18 d0 29 f7  f4
0c59 : 8d 18 d0 60 ac 8d 02 f0  54
0c61 : 07 a0 02 a2 01 4c 85 09  28
0c69 : 4c 76 08 a9 3f 85 3c a9  8c
0c71 : 80 8d 0d 0a a9 0c 8d 0e  8a
0c79 : 0a 20 fa 09 4c 76 08 b1  6f
0c81 : 3b 49 ff 91 3b 4c 0f 0a  f9
0c89 : a9 a0 85 8b a9 98 85 3b  41
0c91 : a9 3e 85 3c 85 8c a9 14  ce
0c99 : 48 a0 07 b1 8b 85 fe b1  6e
0ca1 : 3b aa 0a 8a a2 08 66 fe  07
0ca9 : 2a ca d0 fa 91 3b a5 fe  53
0cb1 : 6a 91 8b 88 10 e5 68 aa  ff
0cb9 : ca f0 1e 8a 48 a5 8b 18  e4
0cc1 : 69 08 85 8b a5 8c 69 00  65
0cc9 : 85 8c a5 3b 38 e9 08 85  63
0cd1 : 3b a5 3c e9 00 85 3c d0  ea
0cd9 : c0 a5 3b 38 e9 a8 85 3b  b2
0ce1 : 85 8b aa a5 3c e9 00 85  a9
0ce9 : 3c 85 8c a8 8a 18 69 08  3f
0cf1 : 85 8b 98 69 00 85 8c c9  81
0cf9 : 1f d0 9b ad 16 d0 29 10  ca
0d01 : f0 23 a5 fb 48 29 f0 4a  92
0d09 : 4a 4a 4a 85 fe 68 29 0f  b2
0d11 : 0a 0a 0a 0a 05 fe 85 fb  3b
0d19 : 20 4d 09 4c 76 08 ad 16  36
0d21 : d0 29 10 f0 03 4c 76 08  24
0d29 : ad 16 d0 09 10 8d 16 d0  9e
0d31 : 20 4d 09 f0 f0 ad 16 d0  cf
0d39 : 29 10 d0 02 f0 e7 ad 16  10
0d41 : d0 29 ef 8d 16 d0 d0 e8  50
0d49 : a9 27 a2 01 a0 08 d0 0f  fa
0d51 : a2 80 ad 16 d0 29 10 f0  da
0d59 : 02 a2 40 a9 00 a0 06 85  1a
0d61 : 8b 86 8c 8c 20 d0 20 bf  6c
0d69 : 0e 20 bf 0e a5 c6 d0 05  17
0d71 : e6 c6 20 b3 ee 78 20 b4  d5
0d79 : e5 c9 9d f0 13 c9 1d f0  9e
0d81 : 3a c9 20 f0 5a c9 a0 f0  1e
0d89 : 59 c9 51 d0 d9 4c 76 08  1f
0d91 : ad 16 d0 29 10 d0 12 a5  be
0d99 : 8b d0 04 a6 8c 30 c7 06  d8
0da1 : 8c 90 c3 26 8c c6 8b 10  78
0da9 : bd a5 8b d0 06 a5 8c c9  89
0db1 : 40 b0 b3 06 8c 06 8c 90  43
0db9 : ad b0 e8 a5 8b c9 27 90  72
0dc1 : 05 a6 8c ca f0 a0 ad 16  8d
0dc9 : d0 29 10 d0 0a 46 8c 90  72
0dd1 : 95 66 8c e6 8b 10 8f 46  9d
0dd9 : 8c 66 8c 10 89 30 f2 a9  f7
0de1 : 00 2c a9 ff 85 fd ad 20  a1
0de9 : d0 29 0f c9 08 f0 5e a9  20
0df1 : 20 85 41 a9 00 85 40 a5  d2
0df9 : 41 85 3c a5 40 85 3b a6  2b
0e01 : 8b f0 17 a0 07 a5 fd 91  97
0e09 : 3b 88 10 fb 18 a9 08 65  c6
0e11 : 3b 85 3b 90 02 e6 3c ca  ce
0e19 : d0 e9 a6 8c ca 8a a2 29  f7
0e21 : a4 fd f0 04 49 ff a2 09  b2
0e29 : a0 07 8d 34 0e 8e 33 0e  75
0e31 : b1 3b 49 00 91 3b 88 10  07
0e39 : f7 18 a5 40 69 40 85 40  dd
0e41 : a5 41 69 01 85 41 c9 3f  09
0e49 : 90 ad 4c 76 08 a9 21 85  ef
0e51 : 41 a9 38 85 40 a5 40 85  63
0e59 : 3b a5 41 85 3c 38 a9 27  e2
0e61 : e5 8b aa f0 19 a0 07 a5  d2
0e69 : fd 91 3b 88 10 fb 38 a5  1c
0e71 : 3b e9 08 85 3b a5 3c e9  f9
0e79 : 00 85 3c ca d0 e7 a5 8c  a0
0e81 : ae 16 d0 e0 10 90 01 0a  28
0e89 : 85 fe 38 e9 01 05 fe a2  52
0e91 : 09 a4 fd d0 04 49 ff a2  56
0e99 : 29 a0 07 8d a5 0e 8e a4  d4
0ea1 : 0e b1 3b 49 00 91 3b 88  0a
0ea9 : 10 f7 18 a5 40 69 40 85  cb
0eb1 : 40 a5 41 69 01 85 41 c9  16
0eb9 : 40 90 9a 4c 76 08 a5 8b  c7
0ec1 : 0a 0a 0a 85 40 a9 00 69  28
0ec9 : 20 85 41 a2 19 a0 07 b1  67
0ed1 : 40 45 8c 91 40 88 10 f7  82
0ed9 : 18 a5 40 69 40 85 40 a5  7e
0ee1 : 41 69 01 85 41 ca 10 e5  3e
0ee9 : 60 a2 20 a0 0a a9 00 f0  86
0ef1 : 06 a2 3e a0 05 a9 07 85  b1
0ef9 : 8b 86 8c 8c 20 d0 20 27  d3
0f01 : 10 20 27 10 a5 c6 d0 05  cb
0f09 : e6 c6 8d 77 02 78 20 b4  72
0f11 : e5 c9 91 f0 13 c9 11 f0  03
0f19 : 3a c9 20 f0 61 c9 a0 f0  27
0f21 : 60 c9 51 d0 d9 4c 76 08  be
0f29 : a5 8c c9 20 d0 04 a5 8b  66
0f31 : f0 cc a5 8b 29 07 f0 0c  09
0f39 : c6 8b a5 8b c9 ff d0 be  fd
0f41 : c6 8c d0 ba a5 8b 38 e9  44
0f49 : 39 85 8b a5 8c e9 01 85  03
0f51 : 8c d0 ab a5 8c c9 3e 90  16
0f59 : 06 a5 8b c9 07 f0 9f a5  10
0f61 : 8b 29 07 49 07 f0 08 e6  52
0f69 : 8b d0 93 e6 8c d0 8f a5  f7
0f71 : 8b 18 69 39 85 8b a5 8c  ee
0f79 : 69 01 85 8c d0 80 a9 00  0d
0f81 : 2c a9 ff 85 3c ad 20 d0  86
0f89 : 29 0f c9 0a f0 72 a5 8c  40
0f91 : c9 3e b0 24 a5 8b 29 f8  77
0f99 : 18 69 41 85 5f a5 8c 69  8f
0fa1 : 01 85 60 a2 40 86 5a ca  08
0fa9 : 86 58 a9 3f 85 5b 85 59  aa
0fb1 : a5 3c 8d 3f 3f 20 bf a3  fb
0fb9 : a9 28 85 40 a5 8b a2 07  2f
0fc1 : 87 3b 29 f8 85 8b ad 20  fb
0fc9 : d0 29 0f aa a5 3c a0 00  06
0fd1 : c4 3b f0 09 e0 05 f0 02  8e
0fd9 : 91 8b c8 d0 f3 91 8b c8  07
0fe1 : c0 08 b0 09 e0 0a f0 02  19
0fe9 : 91 8b c8 d0 f3 a5 8b 69  f9
0ff1 : 07 85 8b a5 8c 69 00 85  71
0ff9 : 8c c6 40 d0 cf 4c 76 08  5c
1001 : a5 8c c9 21 90 b2 85 5b  ee
1009 : a5 8b 29 f8 85 5a a9 20  ef
1011 : 85 60 85 59 a2 01 86 58  50
1019 : ca 86 5f a5 3c 8d ff 1f  21
1021 : 20 71 0a 4c b9 0f a5 8b  c8
1029 : 85 40 a5 8c 85 41 a0 28  fe
1031 : a2 00 a1 40 49 ff 81 40  5f
1039 : 18 a5 40 69 08 85 40 90  30
1041 : 02 e6 41 88 d0 ec 60 09  20
1049 : 0e 20 c8 41 52 44 43 4f  b5
1051 : 50 59 2d cd 41 4b 45 52  7b
1059 : 20 20 20 57 52 49 54 54  e6
1061 : 45 4e 20 42 59 20 c3 48  54
1069 : 52 2e 20 cb 55 52 54 53  34
1071 : 20 20 20 20 20 20 20 20  71
1079 : 20 20 2d 20 c4 49 53 4b  73
1081 : 57 4f 52 4b 2d cd 4f 44  85
1089 : 55 53 0d 0d 00 20 30 0c  47
1091 : a0 00 b9 48 10 f0 06 20  8a
1099 : d2 ff c8 d0 f5 20 cc ff  4a
10a1 : 20 60 a5 86 7a 84 7b a2  2a
10a9 : 07 20 73 00 c9 00 f0 f0  df
10b1 : dd d0 10 f0 08 ca 10 f8  22
10b9 : a2 11 4c e8 10 8a 0a aa  e7
10c1 : bd d8 10 8d ce 10 bd d9  b8
10c9 : 10 8d cf 10 4c 00 00 4e  f7
10d1 : 24 40 51 58 5f 5c 3f 4c  e3
10d9 : 12 2a 11 c2 11 54 08 21  b3
10e1 : 0c b5 12 70 12 55 12 8a  84
10e9 : 0a aa 20 f4 10 20 7a a6  28
10f1 : 4c a1 10 bd 26 a3 85 22  a3
10f9 : bd 27 a3 85 23 a9 00 85  6e
1101 : 13 20 cc ff 20 3f ab a0  43
1109 : 00 b1 22 48 29 7f 20 d2  28
1111 : ff c8 68 10 f4 a0 00 b9  58
1119 : 6a a3 20 d2 ff c8 c0 06  0c
1121 : 90 f5 20 44 12 20 44 12  94
1129 : 60 20 2a 12 90 06 20 44  a8
1131 : 12 4c a1 10 20 44 12 a5  8b
1139 : 7b 85 bc a5 7a 85 bb 20  5e
1141 : 64 12 84 b7 ad e2 03 85  cf
1149 : ba a9 60 85 b9 20 d5 f3  7c
1151 : a5 ba 20 b4 ff a5 b9 20  46
1159 : 96 ff a9 00 85 90 a0 03  bf
1161 : 84 40 20 a5 ff 85 41 a4  3d
1169 : 90 d0 41 20 a5 ff a4 90  c4
1171 : d0 3a a4 40 88 d0 e9 48  d7
1179 : a0 06 20 3f ab 88 d0 fa  45
1181 : 68 a6 41 20 cd bd 20 3f  5a
1189 : ab 20 3f ab 20 a5 ff a6  06
1191 : 90 d0 19 aa f0 06 20 d2  8a
1199 : ff 4c 8d 11 a5 91 10 0c  83
11a1 : ad 8d 02 d0 fb 20 44 12  a5
11a9 : a0 02 d0 b4 20 44 12 a5  cd
11b1 : 90 29 83 d0 06 20 42 f6  29
11b9 : 4c a1 10 20 e0 11 4c a1  e9
11c1 : 10 20 2a 12 b0 0a 20 73  71
11c9 : 00 c9 00 d0 06 20 e0 11  cf
11d1 : 4c a1 10 c9 24 d0 03 4c  99
11d9 : 2a 11 20 09 12 90 f1 a0  64
11e1 : 00 84 90 ad e2 03 85 ba  cf
11e9 : 20 b4 ff a9 6f 85 b9 20  e3
11f1 : 96 ff 20 a5 ff 24 90 70  88
11f9 : 05 20 d2 ff d0 f4 20 ab  4f
1201 : ff 20 44 12 20 44 12 60  91
1209 : ad e2 03 85 ba 20 b1 ff  0c
1211 : a9 6f 85 b9 20 93 ff 20  e9
1219 : 79 00 20 a8 ff 20 73 00  7e
1221 : c9 00 d0 f6 20 ae ff 18  a5
1229 : 60 ad e2 03 85 ba a2 00  32
1231 : 86 90 20 b4 ff 20 ab ff  4e
1239 : a5 90 f0 eb a2 0a 20 f4  c5
1241 : 10 38 60 48 a9 0d 20 d2  b8
1249 : ff 68 60 20 9b b7 8e e2  10
1251 : 03 4c a1 10 20 79 a5 20  89
1259 : 73 00 20 73 00 20 a0 aa  1c
1261 : 4c a1 10 a0 00 b1 7a f0  ef
1269 : 05 c8 c0 28 90 f7 60 20  92
1271 : 2a 12 b0 34 a9 20 8d e0  eb
1279 : 03 a9 00 8d df 03 20 73  80
1281 : 00 a5 7b 85 bc a5 7a 85  d1
1289 : bb 20 64 12 84 b7 ad e2  32
1291 : 03 85 ba a9 00 85 b9 a2  93
1299 : 00 86 0a a0 20 20 d5 ff  cd
12a1 : b0 09 20 44 12 20 e0 11  2e
12a9 : 4c a1 10 20 e8 12 20 e0  2f
12b1 : 11 4c a1 10 20 2a 12 b0  50
12b9 : ef 20 73 00 a5 7b 85 bc  5b
12c1 : a5 7a 85 bb 20 64 12 84  f3
12c9 : b7 a9 00 ae e2 03 a8 20  54
12d1 : ba ff a9 00 85 69 a9 20  80
12d9 : 85 6a a2 40 a0 3f a9 69  c1
12e1 : 20 d8 ff 90 bd b0 c4 a2  39
12e9 : 00 bd f6 12 f0 06 20 d2  2d
12f1 : ff e8 d0 f5 60 0d 4f 50  a3
12f9 : 45 52 41 54 49 4f 4e 20  cb
1301 : 53 54 4f 50 50 45 44 0d  b7
1309 : 00 ad 16 d0 29 10 f0 11  78
1311 : a9 3f 85 3c a9 25 8d 0d  57
1319 : 0a a9 13 8d 0e 0a 20 fa  16
1321 : 09 4c 3f 0d a2 03 b1 3b  41
1329 : 3d 68 13 f0 34 48 3d 6c  d1
1331 : 13 f0 13 68 3d 70 13 d0  d3
1339 : 28 98 4a b0 24 b1 3b 5d  cd
1341 : 68 13 91 3b d0 1b 68 84  8f
1349 : fd 8a 45 fd 4a b0 09 bd  66
1351 : 68 13 11 3b 91 3b d0 09  37
1359 : bd 68 13 49 ff 31 3b 91  d2
1361 : 3b ca 10 c2 4c 0f 0a c0  45
1369 : 30 0c 03 40 10 04 01 80  8e
```

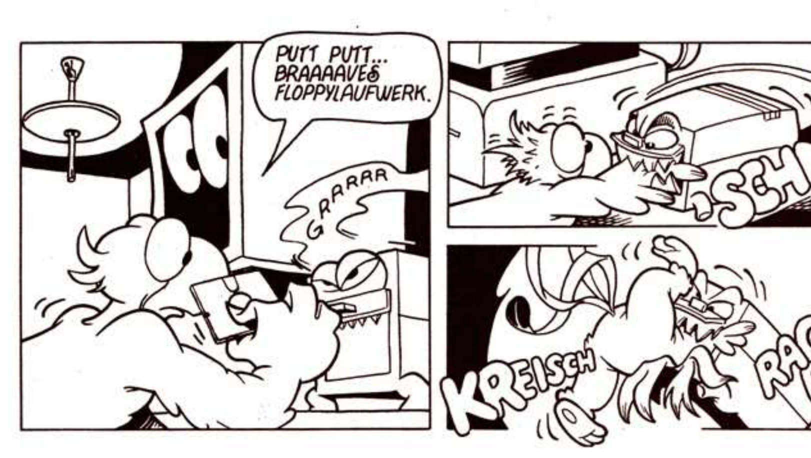
```
1371 : 20 08 02 a9 00 8d 20 d0  da
1379 : 20 a0 13 4c 76 08 ea ea  61
1381 : ea ea ea ea ea ea ea ea  80
1389 : ea ea ea ea ea ea ea ea  88
1391 : ea ea ea ea ea ea 56 f2  4d
1399 : df 2d 00 52 d3 eb e7 a9  e9
13a1 : 20 8d f5 14 a9 04 85 ba  ce
13a9 : a2 00 86 90 86 fe 20 b1  43
13b1 : ff 20 ae ff a6 90 f0 01  21
13b9 : 60 86 b9 86 b7 e8 86 b8  ea
13c1 : 20 c0 ff a6 b8 20 c9 ff  ca
13c9 : a9 ff 85 61 a9 07 8d f6  f6
13d1 : 14 a9 1c 85 97 a9 00 8d  53
13d9 : f1 14 a9 28 8d f3 14 a2  52
13e1 : 04 bd ce 14 20 d2 ff ca  28
13e9 : 10 f7 a9 00 85 63 85 64  b2
13f1 : ad f1 14 85 65 a9 00 8d  0b
13f9 : f7 14 a5 63 a6 64 a4 65  bb
1401 : 20 93 14 ae f7 14 a5 ad  d8
1409 : a0 00 b1 ac ae f7 14 9d  e1
1411 : f8 14 e6 65 e8 8e f7 14  85
1419 : ec f6 14 d0 dd a9 00 a0  0c
1421 : 07 d0 02 d0 b5 ae f6 14  00
1429 : 1e f8 14 2a ca 10 f9 25  6d
1431 : 61 09 80 20 d2 ff ad 8d  3a
1439 : 02 29 01 d0 f9 a5 91 10  5d
1441 : 3d 88 10 e1 a5 63 18 69  ab
1449 : 08 85 63 90 02 e6 64 ce  85
1451 : f3 14 d0 9c a9 0d 20 d2  3f
1459 : ff ad f1 14 18 69 07 8d  32
1461 : f1 14 c6 97 f0 02 d0 bb  db
1469 : a9 04 cd f6 14 f0 0f 8d  87
1471 : f6 14 a9 01 85 97 a9 0f  d6
1479 : 85 61 d0 ea a9 01 85 fe  f7
1481 : a9 0f 20 d2 ff a9 0d 20  d6
1489 : d2 ff 20 cc ff a9 01 4c  e6
1491 : c3 ff 85 14 86 15 98 4a  40
1499 : 4a 4a aa bd d3 14 85 ad  ba
14a1 : 8a 29 03 aa bd ed 14 85  7d
14a9 : ac 98 29 07 18 65 ac 85  37
14b1 : ac a5 14 29 f8 85 63 ad  ff
14b9 : f5 14 18 65 ad 85 ad a5  74
14c1 : ac 18 65 63 85 ac a5 ad  ef
14c9 : 65 15 85 ad 60 50 00 10  79
14d1 : 1b 08 00 01 02 03 05 06  69
14d9 : 07 08 0a 0b 0c 0d 0f 10  4e
14e1 : 11 12 14 15 16 17 19 1a  56
14e9 : 1b 1c 1e 1f 00 40 80 c0  03
14f1 : 00 00 00 00 00 00 00 00  f2
14f9 : 00 00 00 00 00 00 00 80  fb
```

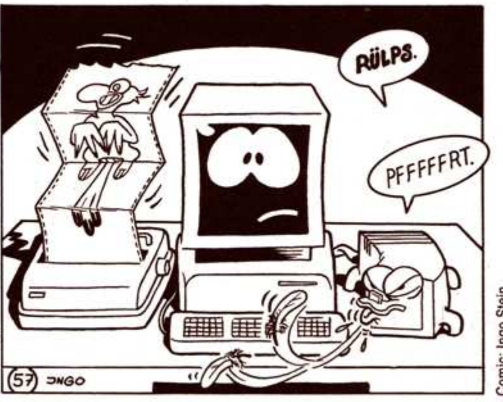
Listing 1. »Hardmaker« (Schluß)

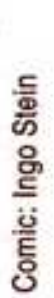
```
Name : multiprint              13a0 171e
-----------------------------------------
13a0 : a9 00 85 b7 ad ea 16 85  37
13a8 : ba 85 b8 ad eb 16 85 b9  02
13b0 : ad 16 d0 29 10 d0 06 a9  b5
13b8 : 81 4c 80 15 ea a6 b8 20  29
13c0 : c0 ff a6 b8 20 c9 ff a9  e4
13c8 : 20 85 15 a0 00 84 14 ae  d6
13d0 : de 03 20 05 14 98 95 57  e4
13d8 : c8 a5 fb 4a 4a 4a 4a aa  30
13e0 : 20 05 14 98 95 57 c8 a5  1d
13e8 : fb 29 0f aa 20 05 14 98  3d
13f0 : 95 57 c8 ae dd 03 20 05  b9
13f8 : 14 98 95 57 20 1a 14 ea  a2
1400 : a9 01 4c c3 ff bd 0a 14  f4
1408 : aa 60 03 00 03 01 02 02  e8
1410 : 03 01 02 03 02 03 02 01  b7
1418 : 02 01 a0 01 b9 44 15 20  35
1420 : 21 15 88 10 f7 a9 00 85  c8
1428 : 62 a9 00 85 63 20 e4 ff  da
1430 : f0 03 4c 0d 15 a0 03 b9  2c
1438 : 40 15 20 21 15 88 10 f7  f5
1440 : a9 00 85 61 a5 14 a4 15  2f
1448 : 85 22 84 23 a4 63 a2 00  54
1450 : 78 a9 35 85 01 b1 22 95  ec
1458 : 64 c8 e8 e0 04 d0 f6 a9  6c
1460 : 37 85 01 58 a9 03 85 60  2f
1468 : a2 03 b5 64 a4 60 f0 06  a3
1470 : 4a 4a 88 4c 6e 14 18 29  c5
1478 : 03 95 5b ca 10 ec a0 00  61
1480 : a2 03 a9 00 85 5f a9 03  0e
1488 : 85 68 b5 5b c5 57 d0 06  80
1490 : b9 48 15 18 90 17 c5 58  3f
1498 : d0 06 b9 54 15 18 90 0d  d3
14a0 : c5 59 d0 06 b9 60 15 18  2a
14a8 : 90 03 b9 6c 15 25 68 05  dc
14b0 : 5f 85 5f 18 26 68 18 26  ff
14b8 : 68 18 ca 10 cd 20 21 15  6d
14c0 : c8 c0 08 d0 bb a6 60 ca  0c
14c8 : 86 60 10 9c 18 a5 22 69  20
14d0 : 08 85 22 a5 23 69 00 85  61
14d8 : 23 a6 61 e8 86 61 e0 28  0b
14e0 : f0 03 4c 4c 14 a6 63 e8  c4
14e8 : e8 e8 e8 86 63 e0 08 f0  8e
14f0 : 03 4c 2d 14 18 a5 14 69  b9
14f8 : 40 85 14 a5 15 69 01 85  60
1500 : 15 a6 62 e8 86 62 e0 19  4f
1508 : f0 03 4c 29 14 a0 01 b9  70
1510 : 46 15 20 21 15 88 10 f7  d3
1518 : 20 cc ff a9 01 20 c3 ff  f3
1520 : 60 a2 01 f0 03 4c d2 ff  0e
1528 : 8d 01 dd a9 10 2c 0d dd  35
1530 : f0 fb ad 00 dd 09 04 8d  db
1538 : 00 dd 29 fb 8d 00 dd 60  02
1540 : 05 00 4c 1b 31 1b 40 1b  df
1548 : 00 00 00 00 00 00 00 00  49
1550 : 00 00 00 00 aa 00 55 00  50
1558 : aa 00 55 00 aa 00 55 00  58
1560 : aa 00 00 00 55 00 00 00  60
1568 : aa 00 00 00 aa 55 aa 55  bd
1570 : aa 55 aa 55 aa 55 aa 55  c5
1578 : 00 00 00 00 00 00 00 00  79
1580 : 85 5b ad 22 15 d0 22 78  b4
1588 : a9 ff 8d 03 dd ad 02 dd  04
1590 : 09 04 8d 02 dd ad 00 dd  46
1598 : 09 04 8d 00 dd a9 10 8d  8d
15a0 : 0d dd ad 0d dd 58 4c c4  04
15a8 : 15 20 cc ff a9 04 ae ea  4c
15b0 : 16 ac eb 16 20 ba ff a9  05
15b8 : 00 20 bd ff 20 c0 ff a2  85
15c0 : 04 20 c9 ff a9 04 85 5e  d4
15c8 : a0 03 20 a2 16 a9 80 85  02
15d0 : 5f a9 02 85 60 a5 5b 29  28
15d8 : c0 d0 04 46 5f 46 60 a5  bf
15e0 : 5b 29 07 a2 00 20 ba 16  fe
15e8 : a5 5b 30 03 4a 4a 4a 29  1a
15f0 : 07 a2 02 20 ba 16 a9 19  02
15f8 : 85 5c a5 5f 30 0a a0 14  ff
1600 : a9 20 20 21 15 88 d0 fa  b4
1608 : a0 08 20 a2 16 a5 5f 20  55
1610 : 21 15 a5 60 20 21 15 a9  e4
1618 : 28 85 5d 78 a9 34 85 01  be
1620 : a0 07 b1 57 99 1e 17 88  93
1628 : 10 f8 a9 37 85 01 58 a5  13
1630 : 57 18 69 08 85 57 90 02  48
1638 : e6 58 a0 08 a2 00 3e 1e  d3
1640 : 17 08 2a e8 28 24 5b 30  74
1648 : 06 e0 08 d0 f1 f0 08 2a  f6
1650 : e4 5e d0 ea 20 21 15 20  94
1658 : 21 15 88 d0 df c6 5d d0  8b
1660 : ba a5 5f 10 26 a2 01 b4  ac
1668 : 57 b5 59 95 57 94 59 ca  b8
1670 : 10 f5 a5 5e 49 0c 85 5e  78
1678 : c9 04 f0 0f a5 5b 30 03  5d
1680 : 4c 17 16 a0 00 20 a2 16  a9
1688 : 4c fa 15 c6 5c d0 f4 a0  d1
1690 : 00 20 a2 16 a0 0d 20 a2  44
1698 : 16 20 cc ff a9 04 20 c3  b4
16a0 : ff 60 b9 ec 16 c9 ff f0  6d
16a8 : 06 20 21 15 c8 d0 f3 60  4d
16b0 : a0 06 d9 96 1f f0 03 88  33
16b8 : d0 f8 a9 00 95 57 a9 20  6a
16c0 : 95 58 60 fb 95 58 60 ae  14
16c8 : e9 16 d0 03 4c d2 ff 48  3d
16d0 : 20 fd 16 ad 00 dd 29 fb  b6
16d8 : 8d 00 dd 09 04 8d 00 dd  66
16e0 : ad 0d dd 29 10 f0 f9 68  f2
16e8 : 60 01 04 01 0d ff ff 1b  f1
16f0 : 31 ff ff ff 1b 4b ff ff  2d
16f8 : ff 1b 32 ff ff 48 ad 0d  24
1700 : dd a9 08 8d 04 dd a9 00  3b
1708 : 8d 05 dd a9 41 8d 0e dd  39
1710 : 68 8d 0c dd 48 ad 0d dd  df
1718 : 29 08 f0 f9 68 60 00 00  4a
```

Listing 2. »MULTIPRINT«, eine Erweiterung des »Hardmaker« für Epson-kompatible Drucker









Super-Hardcopies für Epson-Drucker und Kompatible

Eröffnen Sie sich ganz neue Wege, Grafiken zu Papier zu bringen. Unterschiedliche Größen und unterschiedliche Punktdichten ermöglichen es, extrem kontrastreiche Grafiken zu drucken.

Super-Print ist eine universelle Hardcopyroutine für Hi-Res-Grafiken. Sie ist geschrieben für Epson-Drucker und Kompatible sowie alle grafikfähigen Drucker, die sich über ESC-Sequenzen (CHR$(27)) ansteuern lassen. (Listing 1).

Sie unterstützt alle Grafik-Optionen und bietet darüber hinaus noch vier softwaremäßig erzeugte Dichten sowie Ausgabeformate mit sehr hoher Punktdichte (Bild 1, 2). In der Hardcopy ist keine eigene Centronics-Software integriert, sie muß, falls kein Interface vorhanden ist, vorgeladen werden.

Das Menü

1) bis 6) Druckerparameter

Die Auswahl der Druckerparameter erfolgt mit den Zahlentasten ⟨1⟩ bis ⟨6⟩ oder den Tasten ⟨CURSOR-aufwärts/-abwärts⟩. Mit den Tasten ⟨CURSOR-rechts/-links⟩ können die Parameter verändert werden. Zu den Druckereinstellungen aber später.

P) Print

Mit der P-Taste wird der Ausdruck gestartet. Das Drucken kann jederzeit durch Drücken einer beliebigen Taste unterbrochen werden. Der Drucker wird am Anfang und Ende des Druckvorgangs neu initialisiert.

L) Load

Mit ⟨L⟩ kommt man ins Lademenü. Gibt man nur ⟨RETURN⟩ ein, kommt man wieder ins Hauptmenü. Wird ein »$« als Filename eingegeben, so wird das Inhaltsverzeichnis der Diskette gelistet. Nach »$« können die üblichen Spezifikationen folgen (zum Beispiel $:Nam*). Gibt man nun einen Filenamen ein, gefolgt von ⟨RETURN⟩, so erscheint die Frage »Colorram too? (Y/N)«.

Antwortet man nun mit ⟨N⟩, so wird das File direkt in den Bildspeicher geladen.

Mit ⟨Y⟩ liegt die Ladeadresse $400 Byte tiefer, so daß Files, die zusammen mit dem Farb-RAM gespeichert wurden, ebenfalls korrekt geladen werden.

Achtung!! Super-Print lädt alle Programm-Files, also auch Basic-Programme, in den Grafik-Speicher. Dann ist natürlich in der Regel nur Bit-Müll auf dem Bildschirm zu sehen.

E) Grafik ein

Durch Drücken der E-Taste wird die Grafik eingeschaltet. Anschließend kann mit ⟨R⟩ die Grafik invertiert werden. Sie wird dann auch revers gedruckt.

X) Exit

Mit ⟨X⟩ gelangt man ins Basic zurück. Startet man das Programm wieder mit RUN, so wird dabei der Grafikspeicher neu initialisiert. Eine eventuell vorhandene Grafik wird also gelöscht und muß neu geladen werden.

Super-Print bietet eine Vielzahl von Druckmöglichkeiten, so daß der Umgang am Anfang ein wenig Übung bedarf. Zunächst aber die grundsätzlichen Möglichkeiten, der Schwierigkeit nach geordnet:

6) Sec. Adress

Die Grafik wird generell im Direktmodus zum Drucker geschickt. Da die verschiedenen Interfaces für den Direktmodus unterschiedlicher Sekundäradressen benutzen, lassen sich diese Sekundäradressen zwischen 0 und 80 frei einstellen. Bei den meisten Interfaces ist eine »1« einzusetzen.

5) Linefeed

Es wird bestimmt, ob Super-Print nach dem »CR« (Carriage Return = Wagenrücklauf) am Zeilenende auch noch einen Line-Feed (Zeilenvorschub) senden soll.

4) Left Margin

Hier kann der Abstand vom linken Rand angewählt werden. Der eingestellte Rand leibt auch erhalten, wenn die Grafik über den rechten Blattrand hinausgeht. Der Drucker verschluckt dann den Rest der Zeile, druckt aber trotzdem die Zeilen richtig untereinander, so daß der druckbare Teil des Bildes richtig ausgegeben wird.

1) Size

Es gibt vier verschiedene Ausgabeformate:
Large, Normal, Small und Micro.

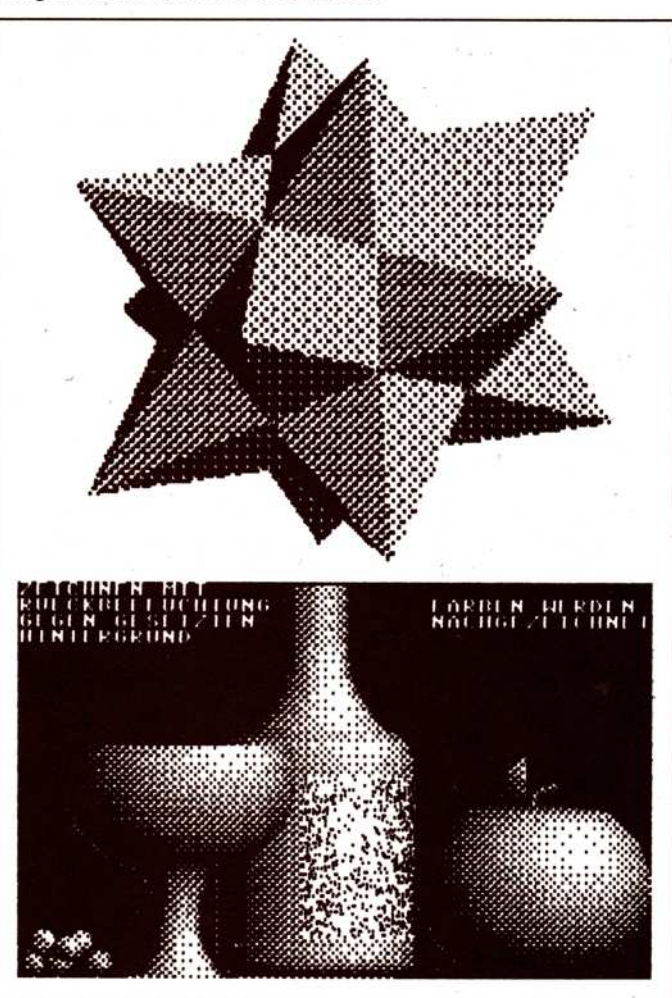


Bild 1 und 2. Ob pechschwarz oder hell, normal oder revers, es gibt nichts, was »Super-Print« nicht kann

In den Modi »Normal« und »Large« beträgt der senkrechte Punktabstand 1/74 Zoll, also normaler Nadelabstand.

Normal: In der Größe »Normal« beträgt damit die Bildhöhe etwa 7 cm. Ein Bildpunkt im Druckbild entspricht dabei einem Punkt auf dem Bildschirm.

Large: Im »Large«-Format wird das Bild doppelt so groß ausgegeben, also einem Bildschirmpunkt entsprechen vier Punkte auf dem Papier mit normaler Dichte. Die Grafik wird dabei um 90 Grad gedreht.

Small: In der Größe »Small« wird der senkrechte Punktabstand auf 2/3 der normalen Höhe reduziert, also 2/216 Zoll. Die Höhe eines Bildschirmausdruckes beträgt dann noch zirka 4,5 cm. In diesem Modus überlappen sich die Bildpunkte im Ausdruck schon ein wenig, es entstehen auch bei normaler Dichte vollschwarze Flächen.

Micro: In der Größe »Micro« wird der senkrechte Punkteabstand weiter reduziert auf 2/216 Zoll. Ein kompletter Bildschirmausdruck ist jetzt noch zirka 2,5 cm hoch. In dieser Auflösung verlaufen die Einzelpunkte zu glatten Konturen. Dieser Modus ist ideal für kleine Embleme etc. in Foto-Qualität.

2) Density

Es stehen drei Dichten zur Verfügung.

Single: druckt das Bild in einfacher Dichte aus.

Double: gibt jeden Punkt zweimal aus, und zwar so, daß zwischen dem ersten und zweiten Mal ein kleiner Versatz entsteht. In den Größen »Large« und »Normal« ist dieser Versatz 2/216 senkrecht, in den Größen »Small« und »Micro« beträgt er 2/270 Zoll waagerecht.

Quadruple steht nur in den Größen Normal und Large zur Verfügung. Hier wird jeder Punkt viermal ausgegeben, und zwar zweimal eng nebeneinander und zweimal eng untereinander. In »Quadruple«-Density werden durchgehende Flächen vollschwarz wiedergegeben, es sind keine Lücken mehr zwischen den Bildpunkten vorhanden.

3) Image Mode

Durch die Option »Size« wird die Höhe der Grafik ausgewählt. Dabei wird aber eine Grafik in »Micro« ebenso breit ausgegeben wie eine in »Normal«, das heißt sie wird sehr breit im Verhältnis zur Höhe. Um diesen Effekt auszugleichen, kann man mit dem Image Mode die Punktdichte des Druckers verändern. Es wird die Punktdichte in »Punkte pro Zeile« angegeben, dahinter in Klammern ein Stichwort.

Die Image Modes werden über die Druckeroption »ESC" * "...« an den Drucker gesendet. Dabei stehen sieben Image Modes zur Verfügung:

0)	Standard Dichte, 480 Punkte/Zeile
1)	Doppelte Dichte, 960 Punkte/Zeile
2)	Doppelte Dichte, Doppelte Geschwindigkeit, 960 Punkte/Zeile
3)	Dreifache Dichte, 1920 Punkte/Zeile
4)	640 Punkte/Zeile
5)	Plotter-Modus, 576 Punkte/Zeile
6)	720 Punkte/Zeile

Falls es sich nicht um einen Epson-Drucker oder Kompatiblen handelt, können die Modi auch etwas anders belegt sein. Um Hardcopies im Bildschirmformat zu erhalten, seien folgende Kombinationen angeraten:

Size		Image Mode
Micro	und	Quadruple
Small	und	Double/Normal Speed
Normal	und	Plotter
Large	und	Plotter

Alle anderen Kombinationen sind natürlich auch möglich. Besondere Aufmerksamkeit erfordert auch der »Plotter«-Mode. Werden Grafiken in den Größen Large und Normal mit dem Image Mode »Plotter« ausgegeben, so ist der Punktabstand senkrecht und waagerecht exakt gleich (Bild 3). Dies ist besonders für den Ausdruck von Platinenvorlagen wichtig.

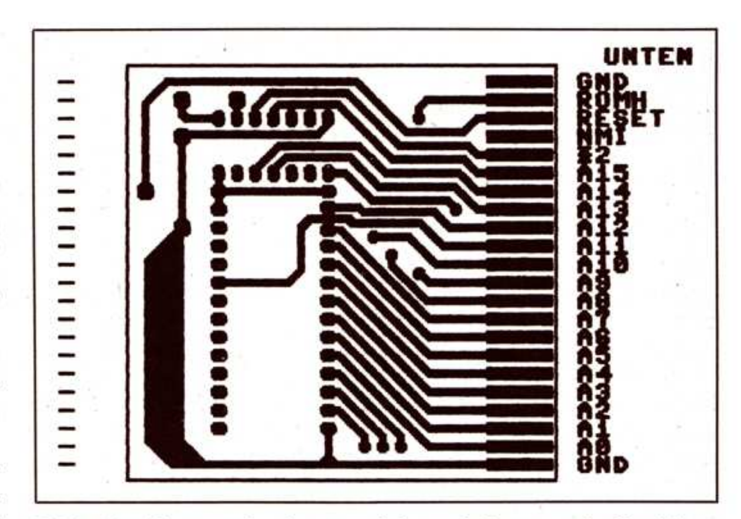


Bild 3. Im Plottmodus lassen sich auch Layouts im Verhältnis 1 : 1 oder 2 : 1 zu Papier bringen

Es sei noch darauf hingewiesen, daß in den Image Modes »Quadruple« und »double Density/double Speed« von zwei direkt nebeneinanderliegenden Punkten nur der linke gedruckt werden kann.

Tips und Tricks

Lademenü

$ beziehungsweise $0:NAME als Filename ruft die Catalog-Funktion auf. Eingabe des Filenamen lädt ein File in den Grafik-Speicher (unabhängig von Anfangsadresse). Akzeptiert auch Grafik-Files mit vorangestelltem Color-RAM (Abfrage).

Speichern der Änderungen

Super-Print mit RUN starten und gewünschte Parameter einstellen. Anschließend mit X ins Basic zurück und wie ein normales Basic-File mit SAVE »NAME« speichern. Super-Print setzt alle Basic-Pointer so, daß der Save-Befehl das Programm vollständig auf die Diskette zurückschreibt.

Grafik-Speicher nicht löschen

Normalerweise wird beim Einsprung in Super-Print der Grafik-Speicher gelöscht. Ist dies nicht erwünscht, folgende Änderung durchführen: Vom Basic aus mit LIST den Programmanfang auslisten,

```
1986 sys 2064+0
```

statt der 0 nun eine 3 einfügen, also

```
1986 sys 2064+3
```

Nun bleibt der Grafik-Speicher auch beim Neustart erhalten. Natürlich kann diese Änderung mitgespeichert werden.

Druckeranpassung

Nach dem Programmstart meldet sich die »Druckeranpassung« (Listing 2) mit der Aufforderung, eine beschreibbare Diskette mit »Super-Print« einzulegen. Es müssen noch mindestens 13 Blocks auf dieser Diskette frei sein.

Zunächst wird nun die Diskette zum Schreiben vorbereitet. Es werden die aktuellen Parameter von Super-Print ausgelesen. Nun wird die Steuersequenztabelle ausgegeben. Dabei erscheint revers der zugehörige Name der Steuersequenz, entsprechend dem Super-Print-Menü.

Anschließend wird der Escape-Code, in der Regel 27, ausgegeben, dann die Steuercodes, zum Beispiel 42 und

0. Falls nur zwei Byte Steuercode benötigt werden, zum Beispiel ESC+K, muß der dritte den Wert 255 erhalten.

Falls der Drucker einige Auflösungen nicht kennt, sollte jeweils die nächstliegende Punktdichte eingestellt werden. Nach den sieben Grafikmodi wird noch »Höchste mögliche Dichte« abgefragt. Hier muß eine der vorher eingegebenen Sequenzen mit hoher Punktdichte noch einmal eingegeben werden, also Quadruple oder, falls nicht vorhanden (zum Beispiel RX80), Double. Es soll auf jeden Fall die höchste Punktdichte eingestellt werden, die der Drucker beherrscht.

Im Anschluß folgt die Steuersequenz, die vor und nach jedem Druckvorgang den Drucker neu initialisiert. Wird keine Initialisierung gewünscht, kann auch dreimal eine 255 eingegeben werden. Nächste Steuersequenz ist die Einstellung des linken Randes (zum Beispiel 27, 108). Super-Print schickt nach dieser Sequenz jeweils noch ein Byte an den Drucker, das den Wert des aktuellen Randabstandes hat. Letzte Sequenz ist die Einstellung des Zeilenvorschubes in n/216 Zoll-Schritten. Die Parameter lassen sich jeweils durch Überschreiben ändern.

Wenn alle Sequenzen editiert sind, werden sie noch einmal zur Kontrolle ausgegeben.

Im nächsten Teil lassen sich die übrigen Parameter einstellen:

— Codes für CR und Linefeed
— Geräteadresse des Druckers und
— die Bildschirmfarben des Menüs

Die Änderungen werden direkt in das Programm übernommen, deshalb ist eine Einstellung in der Regel nur einmal erforderlich.

Sollte das Programm einmal nicht funktionieren, so ist dies kein Grund zur Verzweiflung, es gibt viele mögliche Fehlerquellen, gerade beim Betrieb eines Druckers mit Interface.

Super-Print druckt nicht

Haben Sie einen Drucker am User-Port angeschlossen? Dann wird auf jeden Fall noch ein Centronics-Programm benötigt, Super-Print simuliert keine Schnittstelle. Ein einfaches Centronics-Programm im Bereich $c000 verwenden, da hier keine Störungen zu erwarten sind.

Benutzen Sie ein Interface am seriellen Bus? Jedes Interface benötigt seine speziellen Modifikationen. Meist wird zum Drucken von Grafiken eine besondere Gerätenummer oder Sekundäradresse benötigt (Linear-Kanal).

Haben Sie ein Betriebssystem mit Centronics-Ausgabe im Computer (zum Beispiel Speeddos plus oder ähnliches)? Hier sollte es keine Probleme mit dem Interface geben. Auch hier zuerst testen, ob der Drucker eine besondere Sekundäradresse benötigt.

Wenn nichts geht, zunächst das Druckerhandbuch aufschlagen und das Programm »Druckeranpassung« laden. Stimmen die Steuersequenzen für Grafik-Modi? Ist die Gerätenummer richtig? Super-Print ist eingestellt für Epson FX80 oder Panasonic-Drucker. Andere Drucker brauchen eventuell andere Steuersequenzen.

Nun wieder Super-Print laden und verschiedene Image-Modes testen. Eventuell nochmals die Sekundäradresse ändern.

Und wenn nur Unsinn gedruckt wird?

Dann liegt der Verdacht nahe, daß die Steuersequenzen falsch sind und der Drucker gar nicht in den Grafik-Modus gebracht wird. Wenn allerdings in der Grafik nur einzelne Bits vertauscht sind, kann es sein, daß ein (Soft- oder Hardware-) Interface eine unzulässige Code-Wandlung vornimmt. Ist eventuell durch Ändern der Sekundäradresse zu beheben (Linearkanal). (E. Fiedler/kn)

```
programm : super-print      0801 11
-----------------------------------
0801 : 0e 08 0a 00 9e 20 32 30  aa
0809 : 36 34 aa 30 00 00 00 20  4a
0811 : 70 0d 4c bb 0c 1b 2a 00  d5
0819 : ff 1b 2a 01 ff 1b 2a 02  d6
0821 : ff 1b 2a 03 ff 1b 2a 04  22
0829 : ff 1b 2a 05 ff 1b 2a 06  6f
0831 : ff 1b 2a 03 ff 18 1b 40  56
0839 : ff 1b 6c ff ff 1b 33 ff  86
0841 : ff 0d 0a 04 0f 00 00 00  bb
0849 : 05 0a 00 05 00 00 00 00  f4
0851 : 00 04 ff 00 00 00 00 00  53
0859 : 20 f8 0a ad 47 08 f0 0a  ba
0861 : c9 ff f0 03 4c 8b 09 4c  a4
0869 : f8 08 a9 19 8d 50 08 a9  c2
0871 : 00 8d 52 08 a9 60 8d 53  48
0879 : 08 a9 00 8d 4e 08 a9 28  24
0881 : 8d 51 08 ad 52 08 85 1b  20
0889 : ad 53 08 85 1c 20 53 0b  b9
0891 : a9 40 20 d2 ff a9 01 20  4e
0899 : d2 ff a2 00 38 8a 6a b0  f6
08a1 : 24 aa a0 00 8c 4d 08 8a  ab
08a9 : 31 1b c9 00 f0 09 b9 e5  e4
08b1 : 0e 0d 4d 08 8d 4d 08 c8  8f
08b9 : c0 08 d0 eb ad 4d 08 20  d4
08c1 : d2 ff 18 90 d8 a5 1b 18  02
08c9 : 69 08 85 1b 90 02 e6 1c  e8
08d1 : ce 51 08 d0 c5 20 e4 ff  55
08d9 : f0 03 4c 29 0b 20 64 0b  dc
08e1 : 90 9c a5 1b 8d 52 08 a5  63
08e9 : 1c 8d 53 08 ce 50 08 d0  d3
08f1 : 03 4c 29 0b 4c 7a 08 a9  d2
08f9 : 7e 8d 53 08 a9 00 8d 52  89
0901 : 08 8d 4f 08 a9 28 8d 50  57
0909 : 08 a9 00 8d 4e 08 a9 19  96
0911 : 8d 51 08 ad 53 08 85 1c  c2
0919 : ad 52 08 85 1b 20 53 0b  b8
0921 : a9 90 20 d2 ff a9 01 20  06
0929 : d2 ff a2 07 8a a8 b1 1b  6f
0931 : 2c 4f 08 30 04 4a 4a 4a  5d
0939 : 4a 29 0f a8 b9 ed 0e 20  74
0941 : d2 ff 20 d2 ff ca 10 e4  d6
0949 : a5 1b 38 e9 40 85 1b a5  af
0951 : 1c e9 01 85 1c ce 51 08  e0
0959 : d0 d0 20 e4 ff f0 03 4c  62
0961 : 29 0b 20 64 0b 90 a7 a9  cc
0969 : ff 4d 4f 08 8d 4f 08 d0  f9
0971 : 16 a9 08 18 6d 52 08 8d  06
0979 : 52 08 90 03 ee 53 08 ce  9b
0981 : 50 08 d0 03 4c 29 0b 4c  3d
0989 : 0a 09 a9 19 8d 50 08 a9  74
0991 : 00 8d 52 08 a9 60 8d 53  68
0999 : 08 a9 00 8d 4f 08 a9 28  54
09a1 : 8d 51 08 ad 52 08 85 1b  40
09a9 : ad 53 08 85 1c 20 53 0b  d9
09b1 : a9 40 20 d2 ff a9 01 20  6e
09b9 : d2 ff a2 00 38 8a 6a b0  16
09c1 : 33 aa a0 00 8c 4d 08 8c  de
09c9 : 58 08 a9 80 8d 54 08 ad  97
09d1 : 47 08 c9 02 d0 03 4c 81  28
09d9 : 0a ad 4f 08 c9 02 f0 0f  1d
09e1 : c9 01 f0 03 4c 5c 0a 4e  34
09e9 : 54 08 a0 02 4c 5c 0a a0  bb
09f1 : 01 4c 5c 0a a5 1b 18 69  d7
09f9 : 08 85 1b 90 02 e6 1c ce  02
0a01 : 51 08 f0 03 4c bb 09 20  fa
0a09 : e4 ff f0 03 4c 29 0b ad  1f
0a11 : 48 08 f0 10 ee 4e 08 ad  78
0a19 : 4e 08 c9 01 d0 06 20 87  cb
0a21 : 0b 4c 9f 09 a9 00 8d 4e  c9
0a29 : 08 ee 4f 08 ad 4f 08 c9  86
0a31 : 03 f0 06 20 ac 0b 4c 9f  c6
0a39 : 09 ad 47 08 c9 02 d0 03  e2
0a41 : 4c d2 0a 20 af 0b a5 1b  9d
0a49 : 8d 52 08 a5 1c 8d 53 08  42
0a51 : ce 50 08 d0 03 4c 29 0b  b1
0a59 : 4c 9a 09 8a 31 1b f0 09  48
0a61 : ad 54 08 0d 4d 08 8d 4d  c2
0a69 : 08 4e 54 08 4e 54 08 c8  e8
0a71 : c8 c8 c0 08 30 e5 ad 4d  52
0a79 : 08 20 d2 ff 18 4c be 09  37
0a81 : a5 1b 85 1d a5 1c 85 1e  46
0a89 : ac 4f 08 8a 31 1d f0 09  02
0a91 : ad 54 08 0d 4d 08 8d 4d  f2
0a99 : 08 c8 c8 c8 ad 50 08 c9  62
0aa1 : 01 d0 09 4e 54 08 c0 08  af
0aa9 : 30 e1 10 1b c0 08 30 12  63
0ab1 : 98 38 e9 08 a8 18 a5 1d  fd
0ab9 : 69 40 85 1d a5 1e 69 01  3a
0ac1 : 85 1e 4e 54 08 90 c4 ad  e7
0ac9 : 4d 08 20 d2 ff 18 4c be  ec
0ad1 : 09 20 b2 0b 38 a5 1d e9  f1
0ad9 : 38 8d 52 08 a5 1e e9 01  62
0ae1 : 8d 53 08 ce 50 08 ce 50  15
0ae9 : 08 ce 50 08 ad 50 08 10  0b
0af1 : 03 4c 29 0b 4c 9a 09 20  c4
0af9 : b5 0d a9 02 8d 20 d0 a9  50
0b01 : 04 ae 44 08 ac 4c 08 20  fc
0b09 : ba ff a9 00 20 bd ff 20  5d
0b11 : c0 ff a2 04 20 c9 ff a9  9e
0b19 : 08 20 56 0b a9 09 20 56  38
0b21 : 0b ad 4a 08 20 d2 ff 60  f0
0b29 : a2 00 a9 08 20 56 0b a2  5d
0b31 : 03 20 c9 ff 20 cc ff a9  72
0b39 : 04 20 c3 ff 4c bb 0c 85  1c
0b41 : 1d 86 1e a0 00 b1 1d c9  d3
0b49 : ff f0 06 20 d2 ff c8 d0  38
0b51 : f4 60 ad 49 08 0a 0a a8  54
0b59 : a9 16 85 1d a9 08 85 1e  40
0b61 : 4c 46 0b ad 48 08 f0 4f  70
0b69 : ee 4e 08 2c 48 08 30 0c  a4
0b71 : ad 4e 08 c9 01 d0 3d 20  4c
0b79 : ac 0b 18 60 ae 4e 08 8a  50
0b81 : 29 01 c9 01 d0 18 20 a9  5f
0b89 : 0b a9 07 20 56 0b a9 01  95
0b91 : 20 d2 ff a9 00 20 d2 ff  9c
0b99 : a9 00 20 d2 ff 60 8a c9  65
0ba1 : 02 d0 11 20 ac 0b 18 60  98
0ba9 : a2 00 2c a2 01 2c a2 0e  c3
0bb1 : 2c a2 16 2c a2 17 2c a2  12
0bb9 : 18 ad 3e 08 20 d2 ff ad  2c
0bc1 : 3f 08 c9 ff f0 0d 20 d2  14
0bc9 : ff ad 40 08 c9 ff f0 03  16
0bd1 : 20 d2 ff 8a 20 d2 ff 2c  9c
0bd9 : 4b 08 10 06 ad 43 08 20  43
0be1 : d2 ff ad 42 08 20 d2 ff  33
0be9 : a4 6b 38 60 20 e0 0d ad  f6
0bf1 : 45 08 8d 20 d0 8d 21 d0  41
0bf9 : ad 46 08 8d 86 02 a9 00  9c
0c01 : 8d 11 d0 20 0a 0c 4c cd  1d
0c09 : 0d a9 fd a2 0e 20 40 0b  b7
0c11 : a9 05 8d 54 08 20 1e 0c  3d
0c19 : ce 54 08 d0 f8 ad 54 08  8c
0c21 : 0a aa bd 71 0f 85 1d bd  2b
0c29 : 72 0f 85 1e ad 54 08 20  26
0c31 : b0 0c 20 e0 0d 20 44 0b  04
0c39 : 20 8f 0c ae 54 08 bd 47  05
0c41 : 08 ac 56 08 f0 3a d1 1d  98
0c49 : f0 03 88 d0 f9 98 8d 57  40
0c51 : 08 0a a8 b1 1b 85 1d c8  a2
```

```
0c59 : b1 1b 85 1e 20 44 0b ad  69
0c61 : 55 08 20 b0 0c a9 e3 a2  bb
0c69 : 0f 20 40 0b ad 54 08 20  d8
0c71 : b0 0c a9 f4 a2 0f 20 40  d4
0c79 : 0b ad 54 08 8d 55 08 60  d5
0c81 : aa a9 00 20 cd bd a9 20  b6
0c89 : 20 d2 ff 4c 60 0c ae 54  66
0c91 : 08 bd 0f 10 8d 56 08 8a  fe
0c99 : 0a aa bd 05 10 85 1b bd  1d
0ca1 : 06 10 85 1c bd 15 10 85  64
0ca9 : 1d bd 16 10 85 1e 60 0a  0b
0cb1 : 18 69 04 aa a0 04 18 4c  f7
0cb9 : f0 ff a9 49 a2 11 85 2d  5f
0cc1 : 86 2e 20 ed 0b 20 e4 ff  69
0cc9 : f0 fb c9 58 f0 2e c9 45  66
0cd1 : f0 35 c9 4c f0 2e c9 50  a0
0cd9 : f0 27 c9 9d f0 46 c9 1d  25
0ce1 : f0 5f c9 91 f0 24 c9 11  9f
0ce9 : f0 2d 38 e9 31 30 d6 c9  3f
0cf1 : 06 10 d2 8d 54 08 20 1e  a8
0cf9 : 0c 4c c6 0c a9 93 4c d2  6c
0d01 : ff 4c 59 08 4c e8 0d 4c  57
0d09 : 8c 0d ad 54 08 c9 00 f0  c2
0d11 : b4 ce 54 08 4c f7 0c ad  53
0d19 : 54 08 c9 05 f0 f1 ee 54  87
0d21 : 08 4c f7 0c ad 56 08 f0  5e
0d29 : 0a ad 57 08 f0 97 ce 57  96
0d31 : 08 10 1b ae 54 08 bd 47  e9
0d39 : 08 f0 d4 de 47 08 4c f7  a0
0d41 : 0c ad 56 08 f0 19 cd 57  78
0d49 : 08 f0 c4 ee 57 08 20 8f  2e
0d51 : 0c ac 57 08 b1 1d ae 54  f2
0d59 : 08 9d 47 08 4c f7 0c ae  15
0d61 : 54 08 bd 47 08 c9 50 f0  04
0d69 : a6 fe 47 08 4c f7 0c a0  57
0d71 : 00 84 1b a9 5c 85 1c a2  57
0d79 : 24 a9 01 91 1b c8 d0 fb  18
0d81 : e0 21 d0 01 98 e6 1c ca  0d
0d89 : d0 f1 60 20 b5 0d 20 e4  7c
0d91 : ff f0 fb c9 52 f0 03 4c  92
0d99 : bb 0c a9 60 85 1c a0 00  8c
0da1 : 84 1b a2 20 b1 1b 49 ff  79
0da9 : 91 1b c8 d0 f7 e6 1c ca  d1
0db1 : d0 f2 f0 da 20 e0 0d a9  22
0db9 : 3b 8d 11 d0 ad 18 d0 29  4a
0dc1 : 07 09 78 8d 18 d0 a9 56  78
0dc9 : 8d 00 dd 60 20 e0 0d a9  6a
0dd1 : 1b 8d 11 d0 a9 15 8d 18  bb
0dd9 : d0 a9 97 8d 00 dd 60 ad  e1
0de1 : 12 d0 c9 ff d0 f9 60 a9  7f
0de9 : fb a2 10 20 40 0b 20 ad  76
0df1 : 0e a9 2b a2 11 20 40 0b  1c
0df9 : a5 1b f0 4d a2 49 a0 11  2b
0e01 : 20 bd ff a6 1b a9 24 dd  20
0e09 : 48 11 f0 40 ca d0 f8 a9  88
0e11 : 00 a2 08 a0 00 20 ba ff  64
0e19 : a9 2f a2 11 20 40 0b a9  a8
0e21 : 00 85 cc 20 e4 ff f0 fb  25
0e29 : 85 cc c9 4e f0 07 c9 59  72
0e31 : d0 17 a9 5c 2c a9 60 8d  2f
0e39 : 4d 08 20 70 0d ac 4d 08  1c
0e41 : a2 00 20 b5 0d 8a 20 d5  f3
0e49 : ff 4c bb 0c a9 08 85 ba  45
0e51 : a9 60 85 b9 20 d5 f3 a5  8f
0e59 : ba 20 b4 ff a5 b9 20 96  26
0e61 : ff a9 00 85 90 a0 03 84  09
0e69 : 69 20 a5 ff 85 6b a4 90  b3
0e71 : d0 2f 20 a5 ff a4 90 d0  9f
0e79 : 28 a4 69 88 d0 e9 a6 6b  2c
0e81 : 20 cd bd a9 20 20 d2 ff  7b
0e89 : 20 a5 ff a6 90 d0 12 aa  7e
0e91 : f0 06 20 d2 ff 4c 89 0e  8b
0e99 : a9 0d 20 d2 ff a0 02 d0  da
0ea1 : c6 20 42 f6 20 e4 ff f0  f2
0ea9 : fb 4c e8 0d a9 00 85 1b  8d
0eb1 : 85 cc 20 e4 ff f0 fb c9  4c
0eb9 : 14 f0 18 c9 0d f0 22 c9  f9
0ec1 : 00 30 ef c9 1f 30 eb a6  7f
0ec9 : 1b 9d 49 11 e6 1b 20 d2  95
0ed1 : ff d0 df a6 1b f0 db c6  3b
0ed9 : 1b 78 20 d2 ff 58 4c b3  ee
0ee1 : 0e 85 cc 60 80 40 20 10  9c
0ee9 : 08 04 02 01 00 03 0c 0f  fb
0ef1 : 30 33 3c 3f c0 c3 cc cf  af
0ef9 : f0 f3 fc ff 93 11 20 20  a4
0f01 : 20 20 2a 2a 2a 2a 20 53  1c
0f09 : 55 50 45 52 20 50 52 49  82
0f11 : 4e 54 20 33 2e 30 20 2a  31
0f19 : 2a 2a 2a 0d 11 11 11 11  85
0f21 : 11 11 11 11 11 11 11 11  21
0f29 : 11 11 20 20 20 20 50 29  66
0f31 : 20 50 52 49 4e 54 0d 0d  0d
0f39 : 20 20 20 20 4c 29 20 4c  9d
0f41 : 4f 41 44 0d 0d 20 20 20  76
0f49 : 20 45 29 20 47 52 41 46  f3
0f51 : 49 4b 20 45 49 4e 20 2f  d6
0f59 : 20 52 45 56 45 52 53 20  33
0f61 : 3c 52 3e 0d 0d 20 20 20  8a
0f69 : 20 58 29 20 45 4e 44 ff  dc
0f71 : 7d 0f 8e 0f 9f 0f b0 0f  4f
0f79 : c1 0f d2 0f 31 29 20 53  dc
0f81 : 49 5a 45 20 20 20 20 20  11
0f89 : 20 20 3a 20 ff 32 29 20  c2
0f91 : 44 45 4e 53 49 54 59 20  53
0f99 : 20 20 20 3a 20 ff 33 29  3a
0fa1 : 20 49 4d 41 47 45 20 4d  9b
0fa9 : 4f 44 45 20 3a 20 ff 34  7d
0fb1 : 29 20 4c 45 46 54 20 4d  c8
0fb9 : 41 52 47 49 4e 3a 20 ff  56
0fc1 : 35 29 20 4c 49 4e 45 46  c5
0fc9 : 45 45 44 20 20 20 3a 20  f2
0fd1 : ff 36 29 20 53 45 43 2e  03
0fd9 : 20 41 44 52 45 53 53 3a  a6
0fe1 : 20 ff 11 20 20 20 20 20  0d
0fe9 : 20 20 20 20 20 20 20 20  e9
0ff1 : 20 20 ff 11 a3 a3 a3 a3  70
0ff9 : a3 a3 a3 a3 a3 a3 a3 a3  f8
1001 : a3 a3 a3 ff 2f 10 52 10  3b
1009 : 73 10 00 00 ef 10 03 02  14
1011 : 06 00 01 00 1f 10 23 10  77
1019 : 26 10 00 00 2d 10 02 01  a5
1021 : 00 ff 00 01 ff 00 01 02  49
1029 : 03 04 05 06 ff 00 37 10  2d
1031 : 3d 10 44 10 4b 10 4d 49  86
1039 : 4b 52 4f ff 53 4d 41 4c  be
1041 : 4c 20 ff 4e 4f 52 4d 41  a6
1049 : 4c ff 4c 41 52 47 45 20  85
1051 : ff 58 10 5f 10 69 10 53  a0
1059 : 49 4e 47 4c 45 ff 44 4f  29
1061 : 55 42 4c 45 20 20 20 ff  17
1069 : 51 55 41 44 52 55 50 4c  e7
1071 : 45 ff 81 10 90 10 a0 10  44
1079 : b1 10 c2 10 d3 10 e1 10  4a
1081 : 34 38 30 20 28 53 54 41  d2
1089 : 4e 44 41 52 44 29 ff 39  94
1091 : 36 30 20 28 44 4f 55 42  85
1099 : 4c 45 29 20 20 20 ff 39  4c
10a1 : 36 30 20 28 44 4f 55 42  95
10a9 : 4c 45 2f 44 53 29 20 ff  eb
10b1 : 31 39 32 30 20 28 51 55  45
10b9 : 41 44 52 55 50 4c 45 29  2a
10c1 : ff 36 34 30 20 50 4f 49  43
10c9 : 4e 54 20 20 20 20 20 20  11
10d1 : 20 ff 35 31 32 20 28 50  ca
10d9 : 4c 4f 54 54 45 52 29 ff  f8
10e1 : 37 32 30 20 50 4f 49 4e  83
10e9 : 54 20 20 20 20 ff f3 10  4b
10f1 : f7 10 59 45 53 ff 4e 4f  fc
10f9 : 20 ff 93 11 20 20 20 20  e4
1101 : 20 2a 2a 2a 2a 2a 20 4c  13
1109 : 4f 41 44 20 50 49 43 54  13
1111 : 55 52 45 20 2a 2a 2a 2a  d6
1119 : 2a 0d 0d 0d 0d 0d 20 20  a9
1121 : 20 20 4e 41 4d 45 20 3a  01
1129 : 20 ff 20 0d 0d ff 11 20  48
1131 : 20 20 20 43 4f 4c 4f 52  0b
1139 : 52 41 4d 20 54 4f 4f 20  c0
1141 : 3f 20 3c 59 2f 4e 3e ff  29
```

Listing 1. »Super-Print« bitte mit dem MSE (Seite 159) eingeben

```
10 REM ***********************************    <060>
11 REM *                                 *    <060>
12 REM *    SUPER-PRINT-EDITOR FUER      *    <085>
13 REM *                                 *    <062>
14 REM *       SUPER-PRINT V 3.0         *    <238>
15 REM *                                 *    <064>
16 REM *           11.2.1986             *    <089>
17 REM *                                 *    <066>
18 REM *       ECKEHARD FIEDLER          *    <118>
19 REM *     ADALBERTSTEINWEG 74         *    <237>
20 REM *         5100 AACHEN             *    <249>
21 REM *         0241/500303             *    <134>
22 REM *                                 *    <071>
23 REM ***********************************    <073>
24 :                                          <000>
25 :                                          <001>
90 DIM W(50):WM=49                            <018>
100 PRINT CHR$(147)                           <129>
110 PRINT TAB(5)"**** SUPER-PRINT EDITOR *
    ***"                                      <072>
120 PRINT:PRINT:PRINT:PRINT:PRINT             <133>
130 PRINT TAB(5)"BITTE DISKETTE EINLEGEN U
    ND"                                       <176>
140 PRINT:PRINT                               <092>
150 PRINT TAB(7)"SCHREIBSCHUTZ ENTFERNEN"     <087>
160 PRINT:PRINT                               <112>
180 PRINT TAB(9)"TASTE DRUECKEN > <" CHR$(
    157) CHR$(157);                           <032>
190 POKE 204,0                                <089>
200 GET A$:IF A$=""THEN 200                   <105>
201 :                                         <177>
202 :                                         <178>
203 :  REM ZWISCHENFILE LOESCHEN              <163>
204 :                                         <180>
205 :                                         <181>
210 POKE 204,1:PRINT" "                       <243>
220 PRINT:PRINT                               <172>
230 OPEN 15,8,15,"I"                          <029>
240 NN$ = CHR$(21)+"SUPER-PRINT":REM NAME
    FUER ZWISCHENFILE                         <039>
250 PRINT#15,"S0:"NN$:REM EV. ALTES ZWISCH
    ENFILE                                    <243>
255 :                                         <231>
260 GOSUB 50000:REM LOESCHEN                  <169>
270 IF M1<>1 THEN 60000 :REM ABBRUCH          <022>
290 :                                         <012>
291 :                                         <013>
292 :  REM ALTE WERTE AUSLESEN                <085>
293 :                                         <015>
294 :                                         <016>
300 OPEN 3,8,3,"SUPER-PRINT,P,R"              <005>
310 GOSUB 50000:REM FEHLERMELDUNG HOLEN       <018>
311 PRINT TAB(5)M2$                           <196>
312 IF M1<>0 THEN 60000                       <060>
318 :                                         <040>
319 :                                         <041>
320 FOR I=1 TO 23                             <088>
325 GET#3,A$:IF ST<>0 THEN 60000:REM ABBRU
    CH                                        <042>
330 NEXT I                                    <160>
340 :                                         <062>
341 :                                         <063>
350 FOR I=1 TO WM:REM ALTE WERTE EINLESEN     <052>
360 GET#3,W$: IF ST<>0 THEN 60000:REM ABBR
    UCH                                       <210>
370 W(I)=0:IF W$<>"" THEN W(I)=ASC(W$)        <151>
380 NEXT I                                    <210>
390 CLOSE 3                                   <163>
400 :                                         <122>
```

Listing 2. Druckeranpassung zum Programm »Super-Print«. Bitte mit dem Checksummer (Seite 159) eingeben.

```
401 :                                          <123>
402 :   REM WERTE EDITIEREN                      <198>
403 :                                          <125>
404 :                                          <126>
407 RESTORE: POKE 198,0                        <049>
409 :                                          <131>
410 SM=10:FOR EA=0 TO SM:REM SCHLEIFE DRUC
    KER-CODES <---                             <053>
411 :                                          <133>
412 :                                          <134>
420 PRINT CHR$(147):PRINT                      <157>
430 PRINT TAB(6)"BITTE STEUERSEQUENZEN FUE
    R":PRINT                                   <127>
440 PRINT TAB(10)"DRUCKER EINGEBEN :":PRIN
    T                                          <062>
441 PRINT TAB(2)"(FUER 2-BYTE-BEFEHLE 3. W
    ERT 255!)"                                 <047>
442 PRINT:PRINT:PRINT                          <057>
445 PRINT TAB(2)"STEUERBEFEHL :":PRINT         <159>
449 :                                          <171>
450 READ O$                                    <158>
451 DATA 480 PUNKTE/ZEILE (STANDARD)           <189>
452 DATA 960 PUNKTE/ZEILE (DOUBLE)             <216>
453 DATA 960 PUNKTE/ZEILE DOUBLE SPEED         <127>
454 DATA 1920 PUNKTE/ZEILE (QUADRUPLE)         <029>
455 DATA 640 PUNKTE/ZEILE                      <246>
456 DATA 512 PUNKTE/ZEILE (PLOTTER)            <154>
457 DATA 720 PUNKTE/ZEILE                      <224>
458 DATA HOECHSTE MOEGLICHE DICHTE             <018>
459 DATA DRUCKER INITIALISIEREN                <092>
460 DATA LINKEN RAND EINSTELLEN                <078>
461 DATA ZEILENVORSCHUB N/216                  <007>
462 :                                          <184>
463 PRINT TAB(2)CHR$(18)"*** "O$" ***"         <088>
464 PRINT:PRINT:PRINT                          <079>
465 :                                          <187>
466 :                                          <188>
470 A=4*EA+1                                   <071>
480 B=4*EA+2                                   <087>
490 C=4*EA+3                                   <103>
500 :                                          <222>
510 :                                          <232>
520 PRINT TAB(26)W(A);CHR$(145)                <002>
530 INPUT "{2SPACE}1) ESCAPE-CODE{7SPACE}:
     ";W(A)                                    <070>
535 :                                          <003>
540 PRINT TAB(26)W(B);CHR$(145)                <024>
550 INPUT "{2SPACE}2) ERSTER STEUERCODE :
    ";W(B)                                     <151>
555 :                                          <023>
560 PRINT TAB(26)W(C);CHR$(145)                <046>
570 INPUT "{2SPACE}3) ZWEITER STEUERCODE:
    ";W(C)                                     <081>
575 :                                          <043>
576 :                                          <044>
580 NEXT EA:REM SCHLEIFE DRUCKER-CODES ---
    >                                          <180>
581 :                                          <049>
582 :                                          <050>
583 :   REM DRUCKER-TABELLE OK ?               <103>
584 :                                          <052>
585 :                                          <053>
590 PRINT CHR$(147):RESTORE:REM WERTE DARS
    TELLEN                                     <236>
591 PRINT                                      <185>
592 :                                          <060>
593 FOR I=0 TO SM                              <245>
594 READ A$                                    <192>
595 PRINT A$" :"                               <236>
596 PRINT TAB(24);CHR$(18);W(4*I+1);W(4*I+
    2);W(4*I+3)                                <041>
597 NEXT I                                     <173>
598 :                                          <066>
599 PRINT CHR$(19);                            <196>
600 PRINT TAB(4)"STEUERCODES OK ? [J/N] >
    <"CHR$(157)CHR$(157);                      <177>
610 POKE 204,0                                 <001>
620 GET A$:IF A$=""THEN 620                    <017>
625 POKE 204,1                                 <144>
630 IF A$="N"THEN 400                          <190>
640 IF A$<>"J"AND A$<>"Y"THEN 610              <129>
690 :                                          <158>
691 :                                          <159>
692 :   REM REST EINSTELLEN                    <152>
693 :                                          <161>
694 :                                          <162>
700 PRINT CHR$(147)                            <221>
```

```
705 PRINT:PRINT                                <149>
710 PRINT TAB(5)"**** SONSTIGE PARAMETER *
    ***"                                       <219>
715 PRINT:PRINT:PRINT                          <076>
716 P0=45                                      <144>
718 :                                          <186>
719 :                                          <187>
720 PRINT TAB(25);W(P0);CHR$(145)              <036>
730 INPUT "{2SPACE}CARRIAGE RETURN{6SPACE}
    :";W(P0)                                   <244>
735 PRINT                                      <073>
740 PRINT TAB(25);W(P0+1);CHR$(145)            <047>
750 INPUT "{2SPACE}LINEFEED CHARACTER{3SPA
    CE}:";W(P0+1)                              <078>
755 PRINT                                      <093>
760 PRINT TAB(25);W(P0+2);CHR$(145)            <099>
770 INPUT "{2SPACE}PRINTER-GERAETENUMMER:"
    ;W(P0+2)                                   <189>
775 PRINT                                      <115>
780 PRINT TAB(25);W(P0+3);CHR$(145)            <153>
790 INPUT "{2SPACE}HINTERGRUND-FARBE{4SPAC
    E}:";W(P0+3)                               <176>
795 PRINT                                      <135>
800 PRINT TAB(25);W(P0+4);CHR$(145)            <209>
810 INPUT "{2SPACE}TEXT-FARBE{11SPACE}:";W
    (P0+4)                                     <067>
812 PRINT:PRINT                                <002>
815 :                                          <029>
816 :                                          <030>
817 :   REM PARAMETER OK?                      <205>
818 :                                          <032>
819 :                                          <033>
820 PRINT "{2SPACE}PARAMETER OK ? [J/N/↑]
    > <"CHR$(157)CHR$(157);                    <023>
825 :                                          <039>
830 POKE 204,0                                 <223>
840 GET A$:IF A$=""THEN 840                    <177>
850 POKE 204,1                                 <115>
855 :                                          <069>
860 IF A$="N"THEN 700                          <178>
865 IF A$="↑"THEN 400                          <179>
870 IF A$<>"J"AND A$<>"Y"THEN 830              <153>
875 :                                          <089>
876 :                                          <090>
877 :   REM ABSPEICHERN ?                      <037>
878 :                                          <092>
879 :                                          <093>
880 PRINT CHR$(147)                            <147>
885 PRINT:PRINT:PRINT:PRINT                    <174>
890 PRINT TAB(5)"ABSPEICHERN ? [J/N] > <"C
    HR$(157)CHR$(157);                         <065>
895 :                                          <109>
900 POKE 204,0                                 <037>
910 GET A$:IF A$=""THEN 910                    <086>
920 POKE 204,1                                 <185>
925 :                                          <139>
930 IF A$="N"THEN END                          <154>
940 IF A$<>"J"AND A$<>"Y"THEN 900              <183>
951 :                                          <165>
952 :                                          <166>
953 :   REM SUPER-PRINT-FILE AENDERN           <005>
954 :                                          <168>
955 :                                          <169>
1000 PRINT CHR$(147)                           <011>
1005 PRINT:PRINT:PRINT:PRINT                   <038>
1010 PRINT TAB(10)"WORKING..."CHR$(191)CHR
     $(157);                                   <171>
1020 POKE 204,0                                <157>
1031 :                                         <247>
1032 : REM FILES OEFFNEN                       <147>
1033 :                                         <249>
1100 PRINT#15,"I"                              <101>
1110 OPEN 3,8,3,"SUPER-PRINT,P,R":REM LESE
     FILE                                      <174>
1115 NM$ = NN$+",P,W"                          <042>
1120 OPEN 4,8,4,NM$:REM SCHREIBFILE            <144>
1130 FOR I=1 TO 23:REM ERSTEN 23 BYTES UEB
     ERNEHMEN                                  <216>
1135 GET#3,A$:IF A$=""THEN A$=CHR$(0)          <212>
1140 IF ST<>0 THEN 60000 :REM ABBRUCH          <100>
1150 PRINT#4,A$;                               <025>
1160 NEXT I                                    <228>
1161 :                                         <121>
1162 : REM WERTE EINFUEGEN                     <026>
1163 :                                         <123>
1170 FOR I=1 TO WM:REM ALTE WERTE IGNORIER
     EN                                        <036>
```

```
1175 GET#3,A$:IF ST<>0 THEN 60000:REM ABBR
     UCH                                  <130>
1180 NEXT I                               <248>
1181 :                                    <141>
1182 : REM NEUE WERTE EINFUEGEN           <042>
1183 :                                    <143>
1190 FOR I=1 TO WM                        <112>
1195 PRINT#4,CHR$(W(I));                  <127>
1196 IF ST<>0 THEN 60000:REM ABBRUCH      <156>
1200 NEXT I                               <012>
1201 :                                    <161>
1202 : REM REST UEBERNEHMEN               <110>
1203 :                                    <163>
1210 J=0:FOR I=0 TO 1:REM ENDLOSSCHLEIFE  <169>
1215 :                                    <175>
1220 GET#3,A$                             <133>
1221 :                                    <181>
1225 IF A$=""THEN A$=CHR$(0)              <169>
1230 IF ST<>0 THEN J=1:REM FILE-ENDE ??   <131>
1235 :                                    <195>
1240 PRINT#4,A$;:REM WERT SCHREIBEN       <033>
1245 I=J: NEXT I                          <206>
1246 :                                    <206>
1247 GOSUB 50000:IF M1<>0 THEN 60000      <044>
1250 CLOSE 4: CLOSE 3                     <144>
1255 :                                    <215>
1256 :                                    <216>
1257 :  REM ALTES FILE ERSETZEN           <172>
1258 :                                    <218>
1259 :                                    <219>
1260 PRINT#15,"I"                         <005>
1270 PRINT#15,"S0:SUPER-PRINT":REM ALTES L
     OESCHEN                              <092>
1275 PRINT#15,"I"                         <020>
1280 PRINT#15,"C0:SUPER-PRINT="NN$:REM NEU
     ES AUF ALTEN PLATZ                   <116>
1290 GOSUB 50000                          <032>
1295 PRINT#15,"I"                         <042>
1297 :                                    <003>
1298 :                                    <004>
1300 POKE 204,1:PRINT" "                  <063>
1310 PRINT:PRINT                          <248>
1315 PRINT TAB(5)M2$,M3:REM FEHLERMELDUNG
     ?                                    <228>
1317 :                                    <023>
1318 :                                    <024>
1320 PRINT#15,"S0:"NN$:REM ZWISCHENFILE LO
     ESCHEN                               <049>
1330 GOSUB 50000                          <072>
1335 PRINT#15,"I":REM FLOPPY INITIALISIERE
     N                                    <015>
1337 :                                    <043>
1340 IF M1=1 THEN PRINT TAB(5)"OK":GOTO 14
     00                                   <012>
1350 PRINT TAB(5) M2$                     <219>
1400 CLOSE 15:REM ENDE                    <151>
1500 END                                  <232>
49000 :REM -------------------------------
     ----------------                     <051>
49001 :                                   <208>
49002 :                                   <209>
49003 :  REM FEHLERMELDUNG VON FLOPPY LESE
     N                                    <249>
49004 :                                   <211>
49005 :                                   <212>
50000 INPUT#15,M1,M2$,M3,M4 : RETURN      <066>
59000 :REM -------------------------------
     ----------------                     <145>
59001 :                                   <046>
59002 :                                   <047>
59003 :  REM ABBRUCH BEI DISKETTENFEHLER   <153>
59004 :                                   <049>
59005 :                                   <050>
60000 PRINT:PRINT"DISKETTENFEHLER !!":PRIN
     T                                    <128>
60010 GOSUB 50000: PRINT M2$              <192>
60020 CLOSE 3:CLOSE 4: CLOSE 15:END       <244>
```

Listing 2. Druckeranpassung zu »Super-Print« (Schluß)



Hi-Eddi Druckerroutinen

Vom Star NL-10 bis zum SP 700 VC — Hier finden Sie Druckroutinen für die gängigsten Drucker. Hi-Eddi beinhaltet in seiner Grundversion keinen Druckertreiber für die MPS-Drucker. Hier die Routinen für MPS 801/802/803 und Kompatible.

Hi-Eddi für MPS 802 und Kompatible

Die Druckerroutine für den MPS 802 (Listing 1) hat dieselben Eigenschaften wie die von Hans Haberl geschriebene Routine für Epson-Drucker. Daher kann auf eine Funktionsbeschreibung verzichtet werden. Ich hoffe, daß der Ausdruck von Hi-Eddi-Grafikbildern mit diesem Programm auf allen Versionen des Commodore 1526/MPS 802-Druckers funktioniert. Bei meiner Ausführung (REV. 07C) konnte ich bis jetzt keine Fehler feststellen. (Franz Illetschko/sk)

Hi-Eddi für MPS 801/803 und Kompatible

Eine Lösung für den Commodore MPS 801/803 schickte uns Florian Kulzer (Listing 2). Auch hier braucht am Hi-Eddi selbst nichts verändert zu werden. Die Hardcopy wird mit den Tasten »Commodore« und »P« gestartet.

Allerdings ist mit diesem Programm das Nebeneinanderdrucken zweier Bilder nicht möglich. Ausgegeben wird immer eine Hardcopy des aktuellen Bildschirms. Daher können die Fragen nach Nummer und Format der Hardcopy beliebig beantwortet werden. (Florian Kulzer/sk)

Hi-Eddi auf Star SG-10

Betrifft: Anpassung der Original-Druckerroutine (64'er Ausgabe 1/85, Seite 66) Hi-Print von Hi-Eddi an den SG-10 von Star.

Folgende Änderungen müssen an der Druckerroutine vorgenommen werden:
460 DATA 4: REM Direktkanal
550 DATA 27,51,15,255,255: REM Zeilenabstand $^{15}/_{144}$ Inch
580 DATA 27,103,4,255,255: REM Grafikmodus 4

In dieser Betriebsart werden Kreise in Y-Richtung etwas gedehnt. Wer auf den Ausdruck zweier Bilder nebeneinander und auf Großbilder verzichtet, sollte folgende Werte eingeben. (Zeile 460 muß natürlich trotzdem geändert werden):
550 DATA 27,51,16,255,255: REM Zeilenabstand $^{16}/_{144}$ Inch
580 DATA 27,103,5,255,255: REM Grafikmodus 5

Dafür werden Kreise auch als Kreise dargestellt. Wenn man breiteres Papier als A4 verwendet, kann man auch zwei Bilder nebeneinander und Großbilder drucken. Der Drucker muß auf Star-Betriebsart eingestellt sein (siehe Handbuch). Der User-Port kann weiterhin als Parallelschnittstelle verwendet werden. (Kurt Reiter/sk)

Hi-Eddi für Star NL-10

Die Assemblerroutine »Hi-Print (NL-10)« (Listing 3) wird vom Hauptprogramm Hi-Eddi durch die Tastenkombination <CTRL+P> nachgeladen. Am Hauptprogramm Hi-Eddi sind keine Änderungen erforderlich. Der Druckertreiber ist von der Bedienung und von der Funktion her absolut identisch mit dem Programm Hi-Print (FX-80) aus Ausgabe 1/85 beziehungsweise Sonderheft 6/85. Allerdings besteht keine Möglichkeit, den Drucker über eine Centronics-Schnittstelle am User-Port zu betreiben. Dafür wird mit einer Punktdichte von 1920 Dots/Zeile (beziehungsweise 960 Dots/Zeile für Großbilder) gedruckt. Dadurch ist der Ausdruck proportional zum Bildschirm. Ein Kreis auf dem Bildschirm entspricht exakt einem Kreis auf dem Papier (Bild 1). Ist der Ausdruck abgeschlossen, wird das Programm »Hi-Exe« in Overlay-Technik nachgeladen.

Hi-Eddi für Seikosha GP 700 VC

Auch diese Hi-Print-Version (Listing 4) arbeitet in Overlay-Technik. Folglich ist eine Änderung des Hauptprogramms Hi-Eddi nicht erforderlich. Da der Drucker GP 700 VC eine Farboption hat, wurde eine Routine integriert, die automatisch erkennt, ob Hi-Eddi im Schwarzweiß- oder Farbmodus betrieben wird.

Im Farbmodus ist folgendes zu beachten. Da der C 64 »16«, der Drucker aber nur »8« Farben darstellen kann, wurden die Farbnummern 8 bis 15 denjenigen von 0 bis 7 gleichgesetzt. Daher ist darauf zu achten, daß beim Konstruieren im Farbmodus entsprechende Farben (zum Beispiel Farbe 6 (Blau) und Farbe 14 (Hellblau)) nicht »nebeneinander« vorkommen.

Es lassen sich folgende Formate ausdrucken:
klein —> 320 x 200 Punkte
groß —> 640 x 400 Punkte

Außerdem besteht die Möglichkeit, zwei Bilder im Kleinmodus nebeneinander auszudrucken (640 x 200 Punkte). (W. Wirtz/St. Kirchhoff/sk)

Hi-Eddi mit Itoh 8510

Besitzern des Itoh 8510 bietet dieses Programm die Möglichkeit, von Hi-Eddi Hardcopys zu erzeugen (Bild 2). Da der Itoh 8510 ein verbreiteter und guter Drucker ist, wurden für ihn die im folgenden beschriebenen Programme erstellt. Er wird über die Centronics-Schnittstelle via User-Port am C 64 angeschlossen.

Zusätzlich wurden einige Features eingebaut, die die Epson-Hardcopy-Routine nicht besitzt. So wäre die Vorgabe von Leerspalten vom linken Rand aus (falls noch Platz

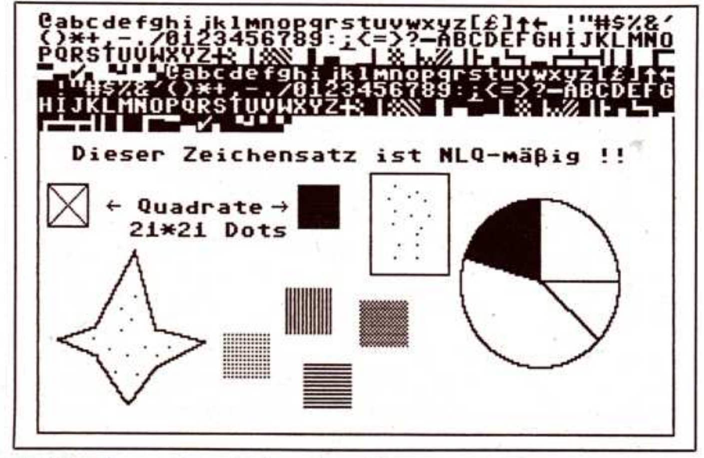


Bild 1. Hardcopy mit dem Star NL-10. Ein Kreis auf dem Bildschirm entspricht exakt einem Kreis auf dem Papier.

ist) eine nützliche Sache. Um zwei Grafiken in der Horizontalen aneinander zu setzen, wurde dazu die volle Bildschirmbreite in 40 Spalten zu je 8 (40 x 8 = 320) Punkten (wie im Textmodus) organisiert. Um jetzt zwei Grafiken nebeneinander zu plazieren, kann dazu die Startspalte (wird mitgedruckt) der linksstehenden Grafik (Grafik Nr. 1) und die Endspalte (wird nicht mitgedruckt) der rechtsstehenden Grafik (Grafik Nr. 2) vorgewählt werden.

Die Ausdruckbreite wird dazu in drei Sektoren von links nach rechts aufgeteilt.

0. Sektor: Abstand (SPC) in Punkten vom linken Papierrand.

1. Sektor: Grafik Nr. 1 (Bild-Nr. ; Startspalte)
   Bild-Nr. = 0 = > kein Bild, Startspalte = 40
   Bild wird rechtsbündig gedruckt
   0 < = Startspalte > = 39
   Startspalte = 0 = > volle Hardcopy von Grafik Nr. 1

2. Sektor: Grafik Nr. 2 (Bild-Nr. ; Endspalte)
   Bild-Nr. = 0 = > kein Bild, Endspalte = 0
   Bild wird linksbündig gedruckt
   1 < = Endspalte < = 40
   Endspalte = 40 = > volle Hardcopy von Grafik Nr. 2

Insgesamt kann folgende Formel für die Aufteilung der drei Sektoren angegeben werden:

SPC + (40—Startspalte) * 8 + Endspalte * 8 < = 640

Im Steuerprogramm werden nacheinander die Bild-Nr. von Sektor 1, dann (falls Bild-Nr. < > 0) die Startspalte von Grafik-Nr. 1, dann die Bild-Nr. von Sektor 2 und (falls Bild-Nr. < > 0) die Endspalte von Grafik-Nr. 2, und schließlich die Anzahl der Leerspalten von Grafik-Nr. 1 abgefragt.

Die Maximalwerte werden stets durch das Steuerprogramm vorgegeben, so daß in vielen Fällen ein einfaches RETURN genügt.

Um die Druckerroutine »HI-P(Itoh8510)« (Listing 5) für den Itoh 8510 in das Steuerprogramm Hi-Eddi einzubinden, mußte dieses auf die Bedürfnisse der Druckerroutine angepaßt werden und heißt nun »HI-E(Itoh8510)«.

Dazu wurden (aus Speicherplatzgründen) alle Zeilen neu numeriert und die Textausgabepassagen verkürzt. In den Programmzeilen 24 und folgende befindet sich das Drucksteuerprogramm.

Es wird hierbei dringend davor gewarnt, das neue Steuerprogramm HI-E(Itoh8510) zu erweitern oder zu ändern, da sonst das Steuerprogramm (inklusive Variablen) nicht mehr in den verfügbaren Basic-Speicherplatz paßt. Dies gilt auch für den eingebauten Parametercheck, da in der Druckroutine selber keine Plausibilitätsprüfung mehr vorgenommen wird.

Für die Initialisierung des Druckers und die Ausgabe der Hardcopy sollte sich der Drucker im SELECT-Modus befinden, da sonst das Programm hängenbleibt (Acknowledge-Signal wird erwartet).

Falls aus irgendeinem Grunde (Drucker beziehungsweise Kabel defekt, falsch oder gar nicht angeschlossen) dieser Fall eintritt, kann mit RUN/STOP-RESTORE die Druckroutine beendet werden. Hi-Eddi kann dann wieder mit RUN gestartet werden. (Das RUN ist notwendig, um HI-EXE nachzuladen, da aus Speicherplatzgründen nach der Anwahl des Drucksteuerprogramms immer die Druckroutine über HI-EXE nachgeladen wird.)

Der Drucker sollte mittels der DIP-Schalter auf CR = CR und 8-Bit-DATA eingestellt sein. Da am Anfang kein Zeilenvorschub ausgegeben wird, sollte man dafür sorgen, daß der Druckkopf sich auf einer freien Zeile befindet. Außerdem sollte das Papier zentriert werden, da sich der Ausdruck über die volle Papierbreite erstrecken kann.

Die Anpassung des Itoh 8510A an den Hi-Eddi besteht im wesentlichen aus zwei Schritten:

1. HI-P(Itoh8510).L (Listing 6) abtippen und starten. Dieses ist ein Ladeprogramm, das die Druckroutine HI-P(Itoh8510) auf Diskette ablegt. Es wird (nach erfolgreichem Test) zum Arbeiten mit Hi-Eddi nicht mehr benötigt.

2. HI-E(Itoh8510) (Listing 5.) abtippen und speichern. Es ist das angepaßte Steuerprogramm und ersetzt das alte Hi-Eddi.

So, das wär's eigentlich. Hi-Eddi mit Itoh 8510A ist nun lauffähig. Mir bleibt nur noch, viel Erfolg mit Hi-Eddi zu wünschen. (Frank Ziesche/sk)

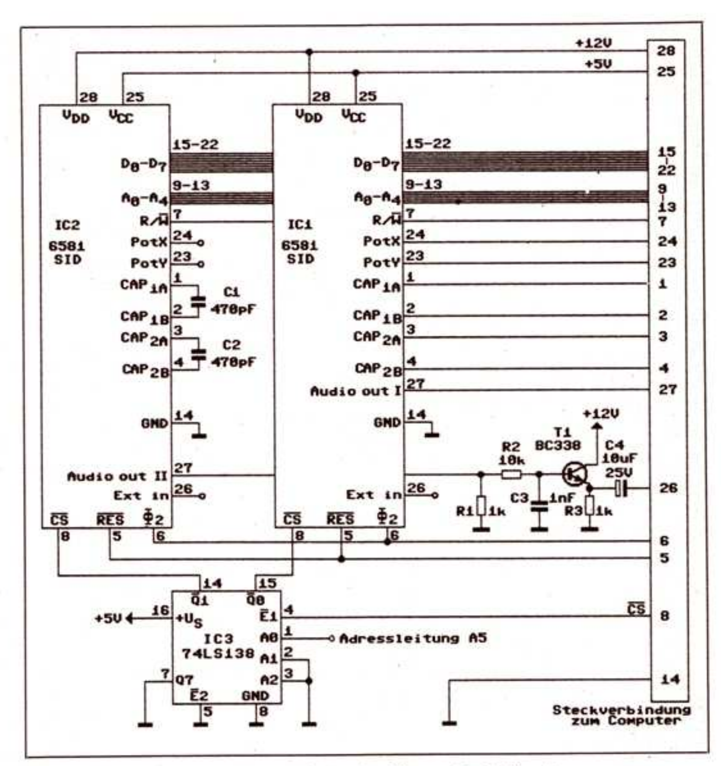


Bild 2. Selbst komplexe Schaltpläne sind für die neuen Druckroutinen kein Problem

Hi-Eddi und Simons Basic

Simons Basic bietet leider keine Möglichkeit, High-Resolution-Bilder auf Diskette zu speichern und zu laden. Dies ist jedoch Voraussetzung, um Bilder, die mit Simons Basic erstellt wurden, auch in Hi-Eddi und andere Programme zu laden.

Ich habe deshalb das Programm »Screensave« (Listing 7) geschrieben, das die Simons-Basic-Befehle »SCRSV« und »SCRLD« um die Fähigkeit, Hires-Bilder zu laden und zu speichern, erweitert. Leider funktioniert das nicht für die Modul-Version, da direkt das Simons Basic modifiziert wird. Dafür wird kein Speicherplatz außerhalb des Simons Basic gebraucht. Außerdem ist die modifizierte Version speicherbar, nach ihrem Laden stehen die erweiterten Befehle sofort zur Verfügung, ohne ein weiteres Programm nachladen zu müssen.

Hinweise zum Eintippen:

1. Simons Basic starten
2. »Screensave« eintippen, sicherheitshalber speichern und starten
3. Nachdem alle DATA-Fehler beseitigt wurden, stehen die erweiterten Befehle »SCRSV« und »SCRLD« zur Verfügung: Die Syntax ist dieselbe wie bisher, mit der Sekundäradresse wird jetzt der Bildschirm gewählt:

Sekundäradresse = 2: Low-Resolution-Bildschirm
Sekundäradresse = 3: High-Resolution-Bildschirm

Das Aufzeichnungsformat des Low-Resolution-Bild-

schirmes ist leider nicht kompatibel mit dem der alten Befehle. Das des High-Resolution-Bildschirmes ist dasselbe wie bei Hi-Eddi, Diashow und anderen Grafikprogrammen. Dazu muß das Bild aber unbedingt als PGM-File, also zum Beispiel mit:
SCRSV1,8,3,"NAME,P,W"
gespeichert werden, da Hi-Eddi keine SEQ-Files laden kann.

Zum Speichern des »neuen« Simons Basic dient das Programm »Saver« (Listing 8). Dieses Programm speichert nicht nur das komplette Simons Basic, das übrigens auch noch andere Modifikationen, zum Beispiel die aus der 64'er Ausgabe 11/84, enthalten kann, sondern auch die Funktionstastenbelegung. Nach dem Laden der mit »Saver« erstellten Simons-Basic-Version stehen somit nicht nur die neuen Befehle, sondern auch die individuelle Funktionstastenbelegung sofort zur Verfügung.

Hinweise zum Eintippen:
1. Funktionstasten nach eigenen Wünschen belegen
2. »Saver« eintippen, speichern, eine Diskette mit mindestens 66 freien Blocks einlegen und Programm starten. Nun fragt der »Saver« nach dem Namen, unter dem er die neue Version ablegen soll. Diese neue Version ist um einen Block länger als die alte (66 statt 65), da die Funktionstastenbelegung hinzugekommen ist.

Tips zu Hi-Eddi

Die Anleitung in den REM-Zeilen des Programms »HI-PRINT.DATA« ist zugegebenermaßen etwas dürftig ausgefallen, so daß eine ausführliche Erklärung dringend nötig ist:

Wie schon in der Bedienungsanleitung erwähnt, beschränkt sich die Anpaßbarkeit dieses Programms auf 8-Nadel-Drucker mit »vernünftiger« Einzelnadelansteuerung, wie zum Beispiel Drucker von Epson oder Mannesmann (im Gegensatz zum 1526/MPS 802). Mit dem Datum in Zeile 420 wird zwischen dem seriellen Bus (also Hardware-Interface) oder dem User-Port gewählt. Im ersten Fall muß mit den Daten in Zeile 440 (Primäradresse) und 460 (Sekundäradresse) das Interface auf Direktmodus gestellt werden. Hardware-Interfaces bieten in der Regel verschiedene Betriebsmodi (zum Beispiel Emulation eines Commodore-Druckers, spezielle List- oder Grafikmodi und eben den Direktmodus oder Linearmodus), die meist über die Sekundäradresse angewählt werden. »HI-PRINT« braucht den Direktmodus, in dem sich das Interface wie ein einfaches Kabel verhält und die Bytes vom Computer unverändert an den Drucker übergibt. Für das Wiesemann-Interface muß zum Beispiel die Sekundäradresse 1 eingegeben werden, für das Görlitz-Interface 4 oder 12 (ohne beziehungsweise mit Auto-Linefeed). Beim Anschluß an den User-Port sind die Daten in Zeile 440 und 460 zwar nicht relevant, man darf sie jedoch nicht weglassen, da sonst die nachfolgenden DATAs verschoben würden.

Aus dem gleichen Grund dürfen auch die Längen der folgenden DATA-Zeilen nicht verändert werden, sie müssen nach Bedarf mit 255 aufgefüllt werden. Außerdem muß jede dieser Zeilen mit mindestens einem 255 enden, da diese Zahl als Ende-Kennzeichen einer Sequenz dient.

Die Zeile 530 legt die Sequenz für ein Carriage-Return Linefeed fest, sie ist 13,255,255 für Drucker mit Auto-Linefeed und 13,10,255 für Drucker ohne Auto-Linefeed.

Zeile 550 stellt den Drucker auf den für Grafik passenden Zeilenvorschub ein. Für den Epson RX-80 ist dies zum Beispiel ESC »3« 23 oder in Zahlen 27,51,23 aufgefüllt mit 255 auf die erforderliche Länge.

In Zeile 580 wird der Grafik-Modus (Bit-Image-Mode) für 640 Punkte/Zeile angesteuert. Beispiele:

In Zeile 660 wird wieder der normale Zeilenabstand eingestellt (ESC »2«) und die -2 in Zeile 610 signalisiert für den DATA-Lader das Ende der DATAs.

Bei Schwarzweiß-Bildern, wie die der Diashow, Supergrafik oder natürlich des erweiterten Simons Basic geht's problemlos. Bei Grafikprogrammen, die im Multicolor-Modus arbeiten, treten zwei Schwierigkeiten auf:
1. Da Hi-Eddi im High-Resolution-Modus arbeitet, sehen Multicolor-Bilder recht ungewöhnlich aus. Dabei kommt es vor allem auch auf die Farborganisation des betreffenden Programms an: Die des Koalapainter ist so chaotisch, daß dessen Bilder im »Hi-Eddi« fast unbrauchbar sind (ein und dieselbe Farbe erhält verschiedene Multicolor-Codes). Paint-Magic-Bilder (siehe Sonderheft 23) sind dagegen sofort »druckreif«.
2. Für Multicolor-Bilder gibt es kein einheitliches Format für die Speicherung auf Diskette, so daß viele Bilder nicht direkt von »Hi-Eddi« geladen werden können.

Laden von Bildern

Für den Koalapainter tritt die zweite Schwierigkeit nicht auf, seine Bilder lassen sich sofort laden. Den Filenamen gibt man dabei in der Form »?PIC...« ein, um das Steuerzeichen am Anfang auszutricksen. Da jedoch Koala-Bilder länger sind als die vom »Hi-Eddi«, werden Farbinformationen nicht nur des geladenen, sondern noch eines weiteren Bildes überschrieben. Nach dem Laden eines Koala-Bildes herrscht somit ein buntes Chaos im Computer, das erst durch Neu-Einfärben beseitigt werden muß.

Etwas komplizierter wird's bei Paint-Magic, dessen Bilder sich nicht direkt laden lassen, da sie in Form eines Basic-Laders abgespeichert werden. Man muß erst ein Bild mit »LOAD"..",8« laden, dann mit RUN starten, anschließend »Hi-Eddi« laden und in Betriebsart 1 starten. Die 1 ist nötig, damit »Hi-Eddi« das im Speicher befindliche Bild nicht löscht. Außerdem darf »Hi-Eddi« nicht im Farbbetrieb gestartet werden, da das Paint-Magic-Bild im Speicherbereich \$4000-\$6000 liegt, wo »Hi-Eddi« sonst keine Farbinformationen ablegen würde. Nach dem Start ist das Bild in Speicher Nummer 7 zu finden.

Lädt man übrigens im Schwarzweiß-Betrieb ein Farbbild (36 Blocks), dann wird auch hier durch die Farbinformation ein Teil eines anderen Bildes überschrieben, was sich durch »Gerümpel« in den ersten paar Zeilen dieses Bildes bemerkbar macht.

Bilder in Basic verwenden

Will man »Hi-Eddi«-Bilder in normalen Basic-Programmen (oder mit der Simons-Basic-Modulversion) laden, dann tut es das Listing »Pic-Lader« (Listing 9), das natürlich in eigene Programme eingebaut werden kann. Das Bild wird unter das Betriebssystem-ROM gelegt, der Farbspeicher ab \$C000 (= 49152, wie bei Simons Basic). Damit wird kein Basic-Speicherplatz belegt. Allerdings dürfen nur Schwarzweiß-Bilder (33 Blocks) geladen werden, bei Farbbildern gibt's einen Absturz.

Auch am Programm selbst sind mir noch vier Kleinigkeiten aufgefallen:
1. Bei einem STOP/RESTORE schreibt das Betriebssystem ein paar Bytes in den RAM-Bereich über \$E000. Diese sind dann in einem der Bilder am unteren Rand als »Gerümpel« zu sehen.
2. Beim Befehl »Paint« kann es bei extrem verwinkelten Flä-

chen vorkommen, daß Teile übersehen werden (Grund dafür ist der begrenzte Stack des 6510). Diese müssen eben extra aufgefüllt werden.

3. Der Sequenzstring sollte nicht länger als zirka 40 Zeichen werden, sonst könnte es bei LOAD oder SAVE zu einem OUT OF MEMORY ERROR kommen. Sollte das passieren, so muß Hi-Eddi mit GOTO 150 gestartet und der Sequenzstring verkürzt werden.

4. Bei gewaltigem »Herumwürgen« am Joystick kann es vorkommen, daß der Computer meint, es würde eine Taste gedrückt. Ergebnis ist dann eine Befehlsausführung, die man gar nicht wollte. Allerdings passiert das äußerst selten, vielleicht liegt's auch nur an meinem mickrigen Joystick. Wenn ansonsten mal ein Befehl nicht das tut, was er sollte, dann liegt es meistens an der eingerasteten SHIFT-LOCK-Taste. Bei Befehlen, die die SHIFT-Taste nicht benutzen, fällt es gar nicht auf. Will man jedoch den Fore-Modus anwählen und färbt statt dessen das ganze Bild ein, dann war's SHIFT-LOCK!

(Hans Haberl/sk)

```
programm : hi-print        0d00 0f9c
------------------------------------

0d00 : 85 5b a9 64 a2 04 a0 00   f7
0d08 : 20 08 0e a9 6a a2 04 a0   f2
0d10 : 06 20 08 0e a2 6a 20 c9   7c
0d18 : ff a9 14 20 d2 ff a2 64   75
0d20 : 20 c9 ff a9 8d 20 d2 ff   7f
0d28 : a9 69 a2 04 a0 05 20 08   72
0d30 : 0e a5 5b 0a 90 03 4c 86   88
0d38 : 0e a5 5b 29 07 a2 02 20   e3
0d40 : 13 0e a9 14 85 fb a5 5b   cd
0d48 : 29 c0 f0 10 a5 5b 4a 4a   02
0d50 : 4a 29 07 a2 00 20 13 0e   ae
0d58 : a9 00 85 fb a9 00 8d 3d   2d
0d60 : 03 20 2d 0e 20 66 0e f0   d0
0d68 : 29 a2 69 20 c9 ff a2 00   68
0d70 : bd 3e 03 20 d2 ff e8 e0   a4
0d78 : 08 d0 f5 a9 0d 20 d2 ff   b8
0d80 : a2 64 20 c9 ff 20 74 0e   84
0d88 : a9 fe 20 d2 ff a9 8d 20   d6
0d90 : d2 ff 18 a5 5e 69 08 85   79
0d98 : 5e 90 02 e6 5f 20 e1 ff   1a
0da0 : f0 3f ee 3d 03 ad 3d 03   2c
0da8 : c9 28 d0 b5 a5 5b 29 c0   cb
0db0 : f0 18 a5 5e a6 5c 85 5c   fe
0db8 : 86 5e a5 5f a6 5d 85 5d   e9
0dc0 : 86 5f a5 fb 49 28 85 fb   c2
0dc8 : d0 92 a2 64 20 c9 ff a9   ba
0dd0 : 0d 20 d2 ff ee 3c 03 ad   da
0dd8 : 3c 03 c9 19 f0 03 4c 5c   3c
0de0 : 0d a9 69 20 c3 ff a2 6a   bc
0de8 : 20 c9 ff a9 24 20 d2 ff   b0
0df0 : a2 64 20 c9 ff a9 8d 20   c9
0df8 : d2 ff a9 6a 20 c3 ff 20   e2
0e00 : cc ff a9 64 20 c3 ff 60   a4
0e08 : 20 ba ff a9 00 20 bd ff   b2
0e10 : 4c c0 ff a0 06 d9 96 1f   98
0e18 : f0 03 88 d0 f8 a9 00 95   ce
0e20 : 5c 8d 3c 03 18 69 20 88   11
0e28 : 10 fb 95 5d 60 78 a9 34   20
0e30 : 85 01 a2 00 8a 9d 3e 03   73
0e38 : e8 e0 08 d0 f8 a9 80 85   96
0e40 : 60 a0 00 b1 5e a2 00 0a   36
0e48 : 90 0a 48 bd 3e 03 05 60   78
0e50 : 9d 3e 03 68 e8 e0 08 d0   32
0e58 : ee 46 60 c8 c0 08 d0 e3   f2
0e60 : a9 37 85 01 58 60 a2 00   39
0e68 : 8a dd 3e 03 d0 05 e8 e0   6b
0e70 : 08 d0 f6 60 18 a5 fb 6d   23
0e78 : 3d 03 aa f0 08 a9 20 20   8e
0e80 : d2 ff ca d0 fa 60 a5 5b   1e
0e88 : 29 07 a2 00 20 13 0e 85   bb
0e90 : 5f a9 00 85 5e 85 fb 85   82
0e98 : 59 85 5a a9 c0 95 fc 4a   c1
0ea0 : 4a e8 e0 04 d0 f7 a9 00   8a
0ea8 : 8d 3d 03 20 2d 0e a5 5a   27
0eb0 : f0 11 a2 00 bd 3e 03 0a   bf
0eb8 : 0a 0a 0a 9d 3e 03 e8 e0   5f
0ec0 : 08 d0 f1 20 62 0f 4c d6   2e
0ec8 : 0e a2 00 bd 4a 03 9d 46   9f
0ed0 : 03 e8 e0 04 d0 f5 20 8d   58
0ed8 : 0f f0 2f a2 69 20 c9 ff   3e
0ee0 : a2 00 bd 46 03 20 d2 ff   37
0ee8 : bd 46 03 20 d2 ff e8 e0   20
0ef0 : 04 d0 ef a9 0d 20 d2 ff   aa
0ef8 : a2 64 20 c9 ff 20 74 0e   fc
0f00 : a9 fe 20 d2 ff a9 8d 20   4e
0f08 : d2 ff 20 e1 ff d0 03 4c   49
0f10 : e1 0d ee 3d 03 a5 59 49   30
0f18 : 08 85 59 d0 ac 18 a5 5e   32
0f20 : 69 08 85 5e 90 02 e6 5f   2e
0f28 : ad 3d 03 c9 50 f0 03 4c   9f
0f30 : ab 0e a2 64 20 c9 ff a9   bb
0f38 : 0d 20 d2 ff a5 5a 49 19   8e
0f40 : 85 5a a5 5e a6 5c 85 5c   44
0f48 : 86 5e a5 5f a6 5d 85 5d   79
0f50 : 86 5f ee 3c 03 ad 3c 03   5e
0f58 : c9 32 f0 03 4c a6 0e 4c   a2
0f60 : e1 0d a9 00 aa a8 9d 46   25
0f68 : 03 e8 e0 08 d0 f8 aa b9   0b
0f70 : 3e 03 0a 90 0b 48 18 b9   8b
0f78 : 46 03 75 fc 99 46 03 68   e5
0f80 : e8 e0 04 d0 ed a2 00 c8   79
0f88 : c0 08 d0 e3 60 a2 00 8a   2d
0f90 : dd 46 03 d0 05 e8 e0 04   8e
0f98 : d0 f6 60 00 ff ff ff ff   fb
```

Listing 1. HI-EDDI und MPS 802. Bitte beachten Sie die Eingabehinweise auf Seite 159

```
programm : hi-print        0d00 0e22
------------------------------------

0d00 : a9 7f a2 04 a0 00 20 ba   92
0d08 : ff 20 c0 ff a2 7f 86 67   56
0d10 : 20 c9 ff a9 ff 85 61 a9   4f
0d18 : 07 85 fd a9 1c 85 97 a9   36
0d20 : 00 8d 12 0e a9 28 8d 14   67
0d28 : 0e a2 04 bd b6 0d 20 d2   3a
0d30 : ff ca 10 f7 a9 00 85 63   0f
0d38 : 85 64 ad 12 0e 85 65 a9   93
0d40 : 00 85 fe a5 63 a6 64 a4   bd
0d48 : 65 20 bb 0d a0 00 b1 ac   78
0d50 : a6 fe 9d 16 0e e6 65 e8   1f
0d58 : 86 fe e4 fd d0 e5 a9 00   39
0d60 : a0 07 a6 fd 1e 16 0e 2a   0c
0d68 : ca 10 f9 25 61 09 80 20   fe
0d70 : d2 ff 88 10 ed a5 63 18   30
0d78 : 69 08 85 63 90 02 e6 64   31
0d80 : ce 14 0e d0 b5 a9 0d 20   13
0d88 : d2 ff ad 12 0e 18 69 07   5d
0d90 : 8d 12 0e c6 97 f0 03 4c   28
0d98 : 24 0d a9 04 c5 fd f0 0c   56
0da0 : 85 fd a9 01 85 97 a9 0f   88
0da8 : 85 61 d0 eb a9 0f 20 d2   c9
0db0 : ff a9 7f 4c cc ff 50 00   fb
0db8 : 10 1b 08 85 14 86 15 98   04
0dc0 : 4a 4a 4a aa bd f4 0d 85   da
0dc8 : ad 8a 29 03 aa bd 0e 0e   52
0dd0 : 85 ac 98 29 07 18 65 ac   17
0dd8 : 85 ac a5 14 29 f8 85 63   d6
0de0 : a9 20 05 ad 85 ad 18 a5   02
0de8 : ac 65 63 85 ac a5 ad 65   4a
0df0 : 15 85 ad 60 00 01 02 03   55
0df8 : 05 06 07 08 0a 0b 0c 0d   07
0e00 : 0f 10 11 12 14 15 16 17   0e
0e08 : 19 1a 1b 1c 1e 1f 00 40   d4
0e10 : 80 c0 a1 00 11 00 00 00   6a
0e18 : 00 80 00 00 00 00 00 ff   58
0e20 : 00 ff 02 c8 cc 00 02 90   af
```

Listing 2. HI-EDDI und MPS 801. Bitte beachten Sie die Eingabehinweise auf Seite 159

```
Name : hi-print            0d00 0e45
------------------------------------
0d00 : 85 5b 20 cc ff a9 04 a2   77
0d08 : 04 a0 00 20 ba ff a9 00   b3
0d10 : 20 bd ff 20 c0 ff a2 04   b1
0d18 : 20 c9 ff a9 04 85 5e a0   79
0d20 : 02 20 13 0e a9 c0 85 5f   2e
0d28 : a9 03 85 60 a9 03 85 f8   7b
0d30 : 24 5b 50 08 a9 80 85 5f   8a
0d38 : a9 07 85 60 24 5b 10 04   38
0d40 : a9 01 85 f8 a5 5b 29 07   d2
0d48 : a2 00 20 21 0e a5 5b 30   f2
0d50 : 03 4a 4a 4a 29 07 a2 02   ae
0d58 : 20 21 0e a9 19 85 5c a5   3c
0d60 : 5b 29 c0 d0 0a a0 14 a9   e3
0d68 : 20 20 d2 ff 88 d0 fa a0   89
0d70 : 06 20 13 0e a5 f8 20 d2   55
0d78 : ff a5 5f 20 d2 ff a5 60   aa
0d80 : 20 d2 ff a9 28 85 5d 78   53
0d88 : a9 34 85 01 a0 07 b1 57   85
0d90 : 99 44 0e 88 10 f8 a9 37   be
0d98 : 85 01 58 a5 57 18 69 08   55
0da0 : 85 57 90 02 e6 58 a0 08   f9
0da8 : a2 00 3e 44 0e 08 2a e8   fe
0db0 : 28 24 5b 30 06 e0 08 d0   f0
0db8 : f1 f0 05 2a e4 5e d0 ea   02
0dc0 : 20 d2 ff 20 d2 ff 20 d2   a0
0dc8 : ff 88 d0 dc c6 5d d0 b7   e5
0dd0 : a5 5b 29 c0 f0 26 a2 01   52
0dd8 : b4 57 b5 59 95 57 94 59   e9
0de0 : ca 10 f5 a5 5e 49 0c 85   50
0de8 : 5e c9 04 f0 0f a5 5b 30   36
0df0 : 03 4c 83 0d a0 00 20 13   4d
0df8 : 0e 4c 5f 0d c6 5c d0 f4   22
0e00 : a0 00 20 13 0e a0 09 20   55
0e08 : 13 0e 20 cc ff a9 04 20   62
0e10 : c3 ff 60 b9 38 0e c9 ff   3d
0e18 : f0 06 20 d2 ff c8 d0 f3   df
0e20 : 60 a0 06 d9 96 1f f0 03   b9
0e28 : 88 d0 f8 a9 00 95 57 18   c6
0e30 : 69 20 88 10 fb 95 58 60   5c
0e38 : 0d ff 1b 33 17 ff 1b 2a   a4
0e40 : ff 1b 32 ff 00 69 00 85   b0
```

Listing 3. Druckertreiber »Hi-Print« für den Star NL-10. Bitte beachten Sie die Eingabehinweise auf Seite 159.

```
Name : hi-print                 0d00 0f11
-----------------------------------------
0d00 : a9 04 aa a0 00 20 ba ff   56
0d08 : a9 00 20 bd ff 20 c0 ff   75
0d10 : a2 04 20 c9 ff ad 0c 03   99
0d18 : 29 40 f0 1e ad 0c 03 4a   3d
0d20 : 4a 4a 29 07 a0 00 20 5f   04
0d28 : 0e a9 01 8d 67 0d 8d 74   fb
0d30 : 0d a9 07 a0 0f a2 01 4c   8a
0d38 : 58 0d a9 00 8d 74 0d 8d   4d
0d40 : 67 0d ad 0c 03 29 80 f0   78
0d48 : 09 a2 02 a9 fd a0 0e 4c   0e
0d50 : 58 0d a2 01 a9 f3 a0 0e   d0
0d58 : 8e 70 0d 8e 8a 0d 20 1e   01
0d60 : ab ad 0c 03 29 07 a0 00   93
0d68 : 20 5f 0e a9 00 85 59 a9   d5
0d70 : 02 85 62 a2 00 86 64 a9   3b
0d78 : 00 85 60 85 61 a6 64 b5   4c
0d80 : 69 8d e3 0d b5 f7 8d 46   28
0d88 : 0e a9 02 85 63 20 c2 0d   f8
0d90 : 20 d2 ff c6 63 d0 f6 e6   58
0d98 : 60 d0 02 e6 61 a5 60 c9   16
0da0 : 40 d0 e6 a5 61 c9 01 d0   c1
0da8 : e0 c6 64 10 ca c6 62 d0   14
0db0 : c2 e6 59 a5 59 c9 c8 d0   99
0db8 : b6 20 cc ff a9 04 20 c3   74
0dc0 : ff 60 a5 59 aa 4a 4a 4a   3f
0dc8 : 0a a8 b9 76 0e 85 f9 b9   cc
0dd0 : 77 0e 85 fa 8a 29 07 18   4d
0dd8 : 65 f9 85 f9 a5 60 29 f8   ce
0de0 : 85 fb a9 20 18 65 fa 85   75
0de8 : 58 18 a5 f9 65 fb 85 57   f0
0df0 : a5 58 65 61 85 58 a5 60   b9
0df8 : 29 07 49 07 aa bd dc 0e   00
0e00 : a0 00 78 a2 34 86 01 31   f1
0e08 : 57 85 fc a2 37 86 01 58   12
0e10 : ae 97 1f f0 0a c9 00 f0   41
0e18 : 03 a9 00 2c a9 01 60 a5   e5
0e20 : 60 85 57 a5 61 4a 66 57   7e
0e28 : 4a 66 57 4a 66 57 85 58   ac
0e30 : a5 59 4a 4a 4a 0a aa bd   79
0e38 : a8 0e 18 65 57 85 57 bd   15
0e40 : a9 0e 65 58 18 69 00 85   2d
0e48 : 58 a0 00 b1 57 a6 fc f0   a7
0e50 : 09 29 f0 4a 4a 4a 4a 4c   2c
0e58 : 5c 0e 29 0f 29 07 60 a2   79
0e60 : 00 dd 96 1f f0 03 e8 10   c3
0e68 : f8 bd e4 0e 99 69 00 bd   9a
0e70 : eb 0e 99 f7 00 60 00 00   cb
0e78 : 40 01 80 02 c0 03 00 05   c7
0e80 : 40 06 80 07 c0 08 00 0a   25
0e88 : 40 0b 80 0c c0 0d 00 0f   82
0e90 : 40 10 80 11 c0 12 00 14   df
0e98 : 40 15 80 16 c0 17 00 19   3d
0ea0 : 40 1a 80 1b c0 1c 00 1e   9a
0ea8 : 00 00 28 00 50 00 78 00   99
0eb0 : a0 00 c8 00 f0 00 18 01   f4
0eb8 : 40 01 68 01 90 01 b8 01   a9
0ec0 : e0 01 08 02 30 02 58 02   dc
0ec8 : 80 02 a8 02 d0 02 f8 02   b9
0ed0 : 20 03 48 03 70 03 98 03   6c
0ed8 : c0 03 e8 03 01 02 04 08   f5
0ee0 : 10 20 40 80 20 40 60 80   a7
0ee8 : a0 c0 e0 40 44 44 48 4c   49
0ef0 : 50 54 58 1b 43 33 32 30   db
0ef8 : 32 30 30 0d 00 1b 43 36   42
0f00 : 34 30 34 30 30 0d 00 1b   01
0f08 : 43 36 34 30 32 30 30 0d   f9
0f10 : 00 a9 13 20 d2 ff a2 25   b0
```

Listing 4. Druckertreiber »Hi-Print« für den Seikosha GP 700 VC. Bitte beachten Sie die Eingabehinweise auf Seite 159.

```
1 IF A<2 THEN A=A+2:LOAD"HI-EXE",8,1                <058>
2 IF A=3 THEN 6                                      <207>
3 IF A=4 THEN 25                                     <035>
4 INPUT"B-ART";C:POKE 780,C:POKE 56,13:CLR
  :W$="123456":SYS 3328                              <117>
5 IF PEEK(8073)AND 64 THEN A=3:LOAD"MENUE"
  ,8,1                                               <238>
6 SYS 3337:A=PEEK(8064):ON A+1 GOSUB 7,10,
  12,11,18,24,11,11,19,20,12:GOTO 6                  <059>
7 OPEN 3,8,0,"$0":GET#3,C$,C$                        <217>
8 GET#3,C$,C$,L$,H$:IF C$=""THEN CLOSE 3:W
  AIT 198,1:RETURN                                   <148>
9 PRINT 256*ASC(H$+CHR$(0))+ASC(L$+CHR$(0)
  );:SYS 3334:GOTO 8                                 <156>
10 PRINT"{CLR,DOWN}SEQ.:":PRINT"{DOWN,2SPA
   CE}"W$:INPUT"{HOME,3DOWN}";W$:IF LEN(W$
   )<2 THEN 10                                       <096>
11 RETURN                                            <069>
12 POKE 8115,8:PRINT"{CLR,DOWN,RVSON}G{RVO
   FF}-BILD":IF PEEK(8073)>127 THEN PRINT"
   {RVSON}F{RVOFF}-BILD"                             <152>
13 PRINT"{RVSON}S{RVOFF}PRITE":PRINT"{RVSO
   N}Z{RVOFF}-SATZ":PRINT"{DOWN}->";                 <208>
14 GET C$:IF C$<>"G"AND C$<>"F"AND C$<>"S"
   AND C$<>"Z"THEN 14                                <093>
15 PRINT C$:POKE 780,ASC(C$):H$=W$:INPUT"{
   DOWN}F-NAME";W$:SYS 3331:W$=H$:H$=""              <213>
16 OPEN 1,8,15,H$:INPUT#1,A,H$:PRINT A;H$:
   IF A>0 THEN WAIT 198,1                            <003>
17 CLOSE 1:RETURN                                    <144>
18 POKE 631,34:POKE 198,1:H$="":INPUT"{CLR
   ,DOWN}";H$:GOTO 16                                <210>
19 A=8075:GOTO 21                                    <059>
20 A=8079                                            <160>
21 D=PEEK(8074):C=PEEK(A+D):PRINT"{CLR,DOW
   N}F"2*D+1:PRINT"S-WEITE:"C,                       <165>
22 INPUT C:IF C<1 OR C>160 THEN 22                   <125>
23 POKE A+D,C:RETURN                                 <180>
24 A=4:LOAD"HI-P(ITOH8510)",8,1                      <215>
25 INPUT"{CLR,DOWN}S1:B-NR.(0=KB)";C:POKE
   3415,C AND 7:IF C=0 THEN D=40:GOTO 27             <236>
26 INPUT"SSP.(<=39;0=VOLL){2SPACE}0{3LEFT}
   ";D:IF D<0 OR D>39 THEN 26                        <222>
27 POKE 3388,D:C=40+D:IF C>40 THEN C=40              <119>
28 INPUT"{DOWN}S2:B-NR.(0=KB)";D:POKE 3435
   ,D AND 7:IF D=0 THEN 30                           <041>
29 PRINT"ESP.(>0;<=";C;"{LEFT}) ";C;:INPUT
   "{5LEFT}";D:IF D<1 OR D>C THEN 29                 <191>
30 POKE 3451,D:C=640-(40-PEEK(3388)+D)*8             <024>
31 PRINT"{DOWN}SPC(=>0;<=";C;"{LEFT}){2SPA
   CE}";C;:INPUT"{6LEFT}";D:IF D<0 OR D>C
   THEN 31                                           <194>
32 C=INT(D/256):POKE 3383,C:INPUT"{DOWN}DR
   UCK{2SPACE}J{3LEFT}";C$:IF C$<>"J"THEN
   34                                                <106>
33 POKE 3381,D-C*256:SYS 3328:INPUT"{DOWN}
   NOCHMAL{2SPACE}J{3LEFT}";C$:IF C$="J"TH
   EN 25                                             <058>
34 A=1:GOTO 1                                        <016>
```


Listing 5. »HI-E(Itoh8510)«. Bitte beachten Sie die Eingabehinweise auf Seite 159.

```
10 REM #### PROGRAMM ZUR ERZEUGUNG #####             <190>
20 REM ## VON HI-P(ITOH8510) AUF DISK ##             <124>
30 REM AUTOR: F.ZIESCHE' AM: 13. 2.85 ##             <068>
40 RESTORE:PS=0:REM DATA'S UEBERPRUEFEN              <136>
50 PRINT"{CLR,DOWN,SPACE}DATA-ZEILEN WERDE
   N UEBERPRUFT !"                                   <100>
60 READ BY : IF BY < 0 THEN 80                       <145>
70 PS = PS + BY : GOTO 60                            <016>
80 IF PS+BY <> 0 THEN PRINT"FEHLER IN DATA
   -ZEILEN !":LIST 160-                              <036>
90 INPUT"{DOWN,SPACE}DISKETTE EINGELEGT{3S
   PACE}JA{4LEFT}";IN$:IF IN$<>"JA"THEN 90           <159>
100 RESTORE : OPEN 3,8,3,"HI-P(ITOH8510),P
    ,W"                                              <061>
110 PRINT#3,CHR$(0);CHR$(13);:REM STARTADR
    ESSE                                             <115>
120 READ BY : IF BY < 0 THEN 140                     <233>
130 PRINT#3,CHR$(BY); : GOTO 120                     <044>
140 CLOSE 3 : PRINT"{DOWN,SPACE}PROGRAMM W
    URDE ERZEUGT !"                                  <107>
150 END                                              <152>
160 REM DATA-ZEILEN FUER HI-P(ITOH8510)              <018>
170 DATA 165,1,72,169,55,133,1,173,87,13,3
    2,186,13,141,170,14,173,107,13,32                <217>
180 DATA 186,13,141,223,14,169,0,141,164,1
    4,141,217,14,141,174,13,32,248,13                <145>
190 DATA 32,212,13,169,27,32,111,14,169,70
    ,32,111,14,169,0,160,0,32,47,14                  <162>
200 DATA 169,0,141,148,14,169,0,141,201,14
    ,169,27,32,111,14,169,83,32,111                  <055>
210 DATA 14,169,0,160,0,32,47,14,169,0,240
    ,16,32,135,14,32,241,14,238,148                  <007>
220 DATA 14,173,148,14,201,40,144,240,169,
    0,240,16,32,188,14,32,241,14,238                 <092>
230 DATA 201,14,173,201,14,201,0,144,240,2
    4,173,164,14,105,64,141,164,14,173               <187>
240 DATA 170,14,105,1,141,170,14,24,173,21
    7,14,105,64,141,217,14,173,223,14                <185>
250 DATA 105,1,141,223,14,169,13,32,111,14
    ,169,10,32,111,14,238,174,13,169                 <197>
260 DATA 0,201,25,176,3,76,42,13,104,133,1
    ,96,162,7,221,149,31,240,4,202,208               <247>
```

```
270 DATA 248,232,138,10,10,10,10,10,96,0,0
    ,0,0,0,0,0,0,24,173,123,13,105,40         <020>
280 DATA 56,237,60,13,133,251,169,0,133,25
    2,160,3,6,251,38,252,136,208,249          <132>
290 DATA 165,251,164,252,141,80,13,140,82,
    13,96,169,255,141,3,221,173,2,221         <055>
300 DATA 9,4,141,2,221,173,0,221,9,4,141,0
    ,221,162,0,189,27,14,32,111,14,232        <033>
310 DATA 224,20,144,245,96,127,27,93,27,78
    ,27,84,49,54,27,102,27,76,48,48           <183>
320 DATA 48,27,48,27,62,133,99,132,98,56,1
    62,144,32,73,188,32,221,189,162           <085>
330 DATA 1,189,0,1,240,6,232,224,6,144,246
    ,202,202,142,81,14,56,169,4,233           <181>
340 DATA 0,240,9,168,169,48,32,111,14,136,
    208,248,172,81,14,240,12,162,1,189        <114>
350 DATA 0,1,32,111,14,232,136,208,246,96,
    141,1,221,173,0,221,41,251,141,0          <177>
360 DATA 221,9,4,141,0,221,173,13,221,41,1
    6,240,249,96,120,165,1,72,41,248          <033>
370 DATA 133,1,169,0,133,252,169,0,133,251
    ,160,3,6,251,38,252,136,208,249           <080>
380 DATA 24,165,251,105,0,133,251,165,252,
    105,0,133,252,177,251,153,204,13          <176>
390 DATA 200,192,8,144,246,104,133,1,88,96
    ,120,165,1,72,41,248,133,1,169,0          <202>
400 DATA 133,252,169,0,133,251,160,3,6,251
    ,38,252,136,208,249,24,165,251,105        <237>
410 DATA 0,133,251,165,252,105,0,133,252,1
    77,251,153,204,13,200,192,8,144           <236>
420 DATA 246,104,133,1,88,96,160,128,140,4
    ,15,169,0,141,8,15,140,10,15,162          <203>
430 DATA 7,189,204,13,41,0,240,7,169,0,9,0
    ,141,8,15,78,10,15,202,16,236,173         <206>
440 DATA 8,15,32,111,14,78,4,15,144,215,96 <148>
450 DATA -57076 :REM  PRUEFSUMME              <184>
```

© 64'er

Listing 6. »HI-P(Itoh8510)« Bitte beachten Sie die Eingabehinweise auf Seite 159.

```
1 REM******      SCREENSAVE      ******      <192>
10 DATA 32,3,178,166,184,32,201,255,152,32
   ,210,255,165,252,32,210,255,169           <217>
20 DATA 210,133,248,162,32,36,250,48,2,162
   ,4,36,250,48,12,177,253,32,0,178          <165>
30 DATA 145,253,177,251,56,176,14,120,169,
   52,133,1,177,251,72,169,54,133            <104>
40 DATA 1,88,104,32,0,178,145,251,200,208,
   29,230,252,230,254,202,208,22,240         <075>
50 DATA 26,234,32,3,178,166,184,32,198,255
   ,32,207,255,32,207,255,169,207            <176>
60 DATA 208,180,234,169,195,36,144,240,183
   ,32,204,255,165,184,32,195,255            <077>
70 DATA 76,42,130,108,248,0,32,219,131,32,
   35,129,160,0,132,251,132,253,165          <107>
80 DATA 185,106,106,133,250,48,8,169,216,1
   33,254,169,4,208,2,169,224,133            <220>
90 DATA 252,169,255,133,249,96,234           <215>
100 S=0:FOR I=45455 TO 45608:READ A:S=S+A:
    POKE I,A:NEXT I                          <253>
110 IF S<>22096 THEN PRINT"FEHLER IN DATA
    S":END                                   <061>
120 PRINT"OK"                                <212>
```

© 64'er

Listing 7. Screensave: Erweiterung für Simons Basic. Bitte beachten Sie die Eingabehinweise auf Seite 159.

```
1 REM******      SAVER      ******           <081>
10 DATA 198,1,169,96,141,123,129,162,2,32,
   201,255,162,97,160,0,185,159,127          <015>
20 DATA 32,210,255,200,202,208,246,169,0,1
   33,253,169,128,133,254,162,64,160         <045>
30 DATA 0,177,253,32,210,255,200,208,248,2
   30,254,202,208,243,185,77,198,32          <008>
40 DATA 210,255,200,208,247,32,204,255,230
   ,1,96,1,8,13,8,191,7,158,40,50            <137>
50 DATA 48,54,51,41,0,0,0,120,169,96,133,2
   52,169,8,133,253,169,0,133,254            <045>
60 DATA 169,128,133,255,162,64,32,69,8,32,
   10,128,169,169,141,123,129,169            <251>
70 DATA 96,133,252,169,72,133,253,169,77,1
   33,254,169,198,133,255,162,1,32           <251>
80 DATA 69,8,76,123,129,160,0,177,252,145,
   254,136,208,249,230,253,230,255           <009>
90 DATA 202,208,242,96,0,0,0,0,0,0,0,0,0 <207>
100 POKE 56,112:CLR                          <079>
110 FOR I=32605 TO 32767:READ A:S=S+A        <000>
120 POKE I,A:NEXT I                          <034>
130 IF S<>21569 THEN PRINT"FEHLER IN DATAS
    ":GOTO 200                               <034>
140 INPUT"NAME DER NEUEN VERSION";N$         <254>
150 OPEN 15,8,15:OPEN 2,8,2,N$+",P,W"        <075>
160 GOSUB 170:SYS 32605:CLOSE 2:GOSUB 170:
    CLOSE 15:GOTO 200                        <197>
170 INPUT#15,A,B$,C,D:PRINT A;B$;C;D         <103>
180 IF A>0 THEN CLOSE 2:CLOSE 15:GOTO 200    <156>
190 RETURN                                   <248>
200 POKE 56,128:CLR:END                      <019>
```

© 64'er

Listing 8. So kann Simons Basic erweitert werden. Bitte achten Sie die Eingabehinweise auf Seite 159.

```
30 REM*  LADEN VON HI-EDDI-BILDERN  *        <222>
100 INPUT"FILENAME";N$                       <042>
110 POKE 781,8:REM  GERAETEADRESSE           <091>
120 POKE 782,0:REM  SEKUNDAERADRESSE         <225>
130 SYS 65466:REM  FILEPARAMETER SETZEN      <203>
140 POKE 780,LEN(N$)                         <181>
150 POKE 781,192:POKE 782,2:REM  POINTER A
    UF FILENAMEN                             <063>
160 FOR I=1 TO LEN(N$):POKE 703+I,ASC(MID$
    (N$,I)):NEXT:REM  FILENAME               <239>
170 SYS 65469:REM  NAMENSPARAMETER SETZEN    <018>
180 POKE 780,0:REM  LOAD-FLAG                <112>
190 POKE 781,0:POKE 782,224:REM  ADRESSE=$
    E000                                     <220>
200 SYS 65493:REM  LOAD                      <083>
210 REM  GRAFIK EINSCHALTEN                  <002>
220 M1=PEEK(56576):POKE 56576,M1 AND 252:R
    EM  VIC-ADRESSBEREICH AB $C000           <212>
230 M2=PEEK(53272):POKE 53272,8:REM  VIDEO
    RAM- UND GRAFIK-ADRESSEN                 <070>
240 M3=PEEK(53265):POKE 53265,M3 OR 32:REM
     HIRES EINSCHALTEN                       <024>
250 F=16*7+6:REM  16*VORDERGRUNDFARBE+HINT
    ERGRUNDFARBE                             <037>
260 FOR I=49152 TO 50152:POKE I,F:NEXT       <138>
270 POKE 198,0:WAIT 198,1:REM  AUF TASTE W
    ARTEN                                    <030>
280 POKE 56576,M1:POKE 53272,M2:POKE 53265
    ,M3:REM  WIEDER NORMALZUSTAND            <030>
```

© 64'er

Listing 9. »PIC.LADER« Hi-Eddi-Bilder in eigenen Programmen verwenden. Bitte mit dem MSE (Seite 159) eingeben.



»Master-Spell« – Das Ende aller Tippfehler

Auch Gutes läßt sich noch verbessern: Master-Text, unser Textverarbeitungs-Star aus dem Sonderheft 16, bekommt ein leistungsfähiges Begleitprogramm, mit dem die Beseitigung von Tippfehlern erheblich vereinfacht wird.

Neben Vizawrite zählt Master-Text, das wir Ihnen erstmals in Ausgabe 6/1986 des 64'er-Magazins und in Version 2.0 in Sonderheft 16 vorstellten, zu den beliebtesten und leistungsstärksten Textprogrammen für den C64. Eine automatische Hilfe zur Überprüfung der Rechtschreibung fehlte aber bislang. Mit dem hier vorgestellten Programm »MASTER-SPELL« (Listing 1) brauchen nun jedoch auch Benutzer von Master-Text 64 auf einen solchen Spell-Checker nicht länger zu verzichten. »MASTER-SPELL« ist ähnlich aufgebaut wie das ebenfalls in diesem Sonderheft enthaltene Programm »VIZA.SPELLER« (Seite 61) und verwendet sogar dasselbe Wortverzeichnis, so daß Sie Ihre individuellen Verzeichnisse bei einem Wechsel von einem zum anderen Programm weiterverwenden können. Diese alphabetische Wörterliste wird mit dem Hilfsprogramm »DICTIONARY.GEN« (Listing 5 des Beitrages über die Vizawrite-Werkzeuge) erzeugt und ist zunächst leer, läßt sich aber mit Hilfe vorhandener Texte leicht auffüllen und an individuelle Bedürfnisse anpassen. Auf der Programmservice-Diskette befindet sich bereits eine solche Sammlung von deutschen Wörtern, die als Ausgangsbasis für ein umfangreiches elektronisches Wörterbuch dienen kann.

Die Arbeitsweise von »MASTER-SPELL« ist im Prinzip recht einfach. Vor dem eigentlichen »Spell-Checking« wird von dem zu prüfenden Text erst einmal eine alphabetische Wortliste erzeugt. Diese Wortliste vergleicht das Programm anschließend mit seinem Wörterbuch. Unbekannte, das heißt nicht im Verzeichnis vorhandene Wörter können in das Wörterbuch aufgenommen, ignoriert oder – falls es sich um falsch geschriebene Wörter handelt – für einen spätere Korrektur vorgemerkt werden. Dieser letzte Schritt muß von Hand vorgenommen werden, ist aber nur eine kleine Mühe, zumal dann, wenn sich nicht sehr viele falsche Wörter im Text befinden.

»MASTER-SPELL« wird unabhängig von Master-Text wie ein Basic-Programm geladen und gestartet. Nach dem Start erscheint in der zweiten Bildschirmzeile ein kleines Menü, bestehend aus den Buchstaben »c«, »d«, »e«, »f«, »i«, »l«, »u«, »w«, »C«, »M« und »P«. Diese einzelnen Menüpunkte werden durch Druck auf die entsprechende Taste aufge-

rufen. Hinter den Buchstaben verbergen sich dabei folgende Funktionen:

c – Check Wortliste

Dies ist die eigentliche Check-Funktion, damit wird die Überprüfung der im Speicher befindlichen Wortliste des aktuellen Textes veranlaßt. Voraussetzung hierfür ist also, daß zunächst über den Menüpunkt »l« ein mit Master-Text geschriebener Text geladen wurde, ferner, daß sich im Diskettenlaufwerk eine Diskette mit einer Wortschatz-Bibliothek befindet. Wird versehentlich ein falscher Filename eingegeben, der kein Textfile repräsentiert, so wird dies vom Programm erkannt und mit einer Fehlermeldung quittiert.

Sollte während des Korrekturvorganges ein unbekanntes, also (noch) nicht in der Wortschatzliste vorhandenes Wort gefunden werden, so unterbricht das Programm die Suche, und es erscheint eine entsprechende Mitteilung auf dem Bildschirm. Sie können dann entscheiden, ob das unbekannte Wort »gelernt«, also in die Liste aufgenommen, oder übergangen werden soll. Drücken Sie dazu <J> oder <N>. Wollen Sie den Korrekturvorgang jedoch vorzeitig abbrechen, so ist dies mit <RUN/STOP> an diesen Haltepunkten möglich.

d – Directory

Es wird das Inhaltsverzeichnis der eingelegten Diskette angezeigt.

e – Wörter löschen

Wird ein Wort fälschlicherweise übernommen, oder wird es aus anderen Gründen nicht mehr benötigt (etwa ein nicht mehr benutzter Spezialausdruck oder Name), so kann es mit dieser Funktion wieder aus der Wortschatzliste gestrichen werden. Dazu geben Sie das jeweilige Wort oder zumindest den Wortanfang ein. Der Computer durchsucht nun den gespeicherten Wortschatz und gibt jedes Wort, das mit der gesuchten Zeichenfolge beginnt, auf dem Bildschirm aus. Dabei können Sie sich dann jeweils entscheiden, ob Sie das Wort löschen wollen oder nicht.

f – Suchen eines Wortes

Wollen Sie wissen, welche Wörter in Ihrem Verzeichnis enthalten sind, oder wollen Sie die Schreibweise eines bestimmten Ausdrucks nachsehen, so geben Sie bei dieser Funktion wie bei »e« den Anfang des gesuchten Wortes ein. Das Programm durchsucht dann die Datei nach Wörtern, die mit der entsprechenden Zeichenfolge beginnen und listet diese am Bildschirm auf. Die deutschen Umlaute werden selbstverständlich am Bildschirm angezeigt. Für die Suche nach den Sonderzeichen gelten folgende Regeln:

Gesuchtes Zeichen	Eingabe auf der Tastatur
ä	[(linke eckige Klammer)
ü	] (rechte eckige Klammer)
ö	£ (Pfund-Zeichen)
ß	↑ (Pfeil-oben-Zeichen)

Die Eingabe der Umlaute muß im übrigen auch beim Menüpunkt »e« auf die gleiche Art vorgenommen werden.

i – Wort-Statistik

Es wird die Gesamtzahl aller Wörter im Text sowie die Zahl der unterschiedlichen Wörter angegeben. Hier ein Tip:

Wer an seinem Schreibstil arbeiten will, hat mit dieser Funktion ein vorzügliches Hilfsmittel, um sogenannte »Lieblingswörter« zu identifizieren.

l – Textfile laden

Dieser Menüpunkt muß zu Beginn eines Korrekturvorganges angewählt werden. Geben Sie dabei den Namen des gewünschten Textfiles bitte ohne die bei Master-Text übliche Kennung »t« an. Nach dem Laden wird automatisch die für die Rechtschreibüberprüfung benötigte alphabetische Wortliste des Textes erstellt.

u – Unbekannte Wörter ausgeben

Nachdem Sie Ihr Dokument mit »c« oder »C« überprüft haben, können Sie mit diesem Menüpunkt die unbekannten Wörter auf dem Bildschirm auflisten.

w – Ausgabe der Wortliste

Dieser Menüpunkt zeigt Ihnen die Wortliste des jeweiligen Textes auf dem Bildschirm (Bild 1).

Da bei diesem Programm Wert auf eine möglichst große Wortspeicherkapazität gelegt wurde, können – wie schon erwähnt – nur die einzelnen Wörter automatisch auf korrekte Schreibung überprüft werden. Die im Text befindlichen



Bild 1. Eine von »MASTER-SPELL« erzeugte Wortliste

falschen Wörter müssen Sie aber anschließend von Hand korrigieren. Dazu gehen Sie folgendermaßen vor: Sie rufen nach dem Überprüfen der Wörter die Wortliste des Textes mit »w« auf. Alle Wörter, die nicht im Wortschatz vorgekommen sind, werden jetzt in der Liste invers dargestellt (Bild 1). Kontrollieren Sie nun einfach, welche dieser Wörter falsch geschrieben sind. Anschließend rufen Sie wie gewohnt Master-Text auf und laden den soeben überprüften Text erneut. Sie können nun von dort aus mit <CTRL S> nach den fehlerhaften Wörtern suchen und die endgültige Korrektur durchführen.

C – Wortliste überprüfen

Dieser Menüpunkt (<SHIFT C>) überprüft die Wortliste, ohne daß Sie dabei am Computer verweilen müssen. Es wird nur der Inhalt der Wortliste überprüft und die unbekannten Wörter markiert. Es erfolgt keine Abfrage, ob diese Wörter gelernt werden sollen oder nicht.

M – unbekannte Wörter zufügen

Nach Überprüfung der Wortliste können Sie diese mit »M« in das Wörterverzeichnis einfügen.

P – Ausdruck unbekannter Wörter

Unbekannte Wörter können auf dem Drucker ausgegeben werden, um sie später im Text zu verbessern. Sie müssen darauf achten, daß sich der Drucker im Kleinschriftmodus befindet. Umlaute werden sonst nicht korrekt ausgedruckt werden.

Abschließend noch ein Hinweis: Ein Spell-Checker ist nur so gut, wie das Wörterverzeichnis, das er zur Kontrolle verwendet. Achten Sie deshalb sorgfältig darauf, daß Sie sich systematisch einen für Ihre Zwecke geeigneten Wortschatz aufbauen. Erst danach kommen Sie in den vollen Genuß von »MASTER-SPELL«. Und noch etwas: Ein Spell-Checker überprüft lediglich die korrekte Schreibweise einzelner Wörter. Ob die Wörter an der jeweiligen Textstelle grammatikalisch, stilistisch oder sachlich richtig verwendet wurden, erkennt er nicht. Auch der beste Spell-Checker ist darum nur eine Hilfe für eindeutige Schreibfehler und erspart Ihnen nicht, Ihre Texte mit Verstand zu überprüfen.

(Dieter Bayer/Dr. Rudolf Egg)

```
Name : master-spell       0801 19e7
-----------------------------------
0801 : 0d 08 c3 07 9e 28 32 30   38
0809 : 36 33 29 00 00 00 20 80   a5
0811 : 08 20 97 17 a9 00 85 c6   30
0819 : 20 e4 ff a2 0a dd 5f 08   1d
0821 : f0 05 ca 10 f8 30 f1 8a   36
0829 : 0a aa bd 6a 08 8d 49 08   67
0831 : bd 6b 08 8d 4a 08 8e c3   fe
0839 : 19 a0 03 bd 28 04 09 80   e3
0841 : 9d 28 04 e8 88 d0 f4 20   34
0849 : ff ff ae c3 19 a0 03 bd   8a
0851 : 28 04 29 7f 9d 28 04 e8   b3
0859 : 88 d0 f4 4c 12 08 43 44   07
0861 : 45 46 49 4c 55 57 c3 cd   60
0869 : d0 47 10 49 15 25 14 56   82
0871 : 13 34 0e 38 09 b7 0f 03   ba
0879 : 0e 16 0f b2 0f cc 0f a9   93
0881 : 0b 8d 20 d0 a9 0b 8d 21   e0
0889 : d0 a9 0c 8d 86 02 a2 00   e6
0891 : 9d 00 d8 9d 00 d9 9d 00   5d
0899 : da 9d 00 db ca d0 f1 a9   0c
08a1 : 80 8d 8a 02 8d 00 28 20   84
08a9 : ba 08 a9 18 8d 18 d0 a9   05
08b1 : ce a0 17 20 ae 14 4c 93   7d
08b9 : 0d 78 a9 33 85 01 a0 00   b6
08c1 : 84 50 84 52 a9 d8 85 51   f3
08c9 : a9 20 85 53 a2 08 b1 50   20
08d1 : 91 52 88 d0 f9 e6 51 e6   b1
08d9 : 53 ca d0 f2 a2 17 bd 00   fe
08e1 : 09 9d d8 20 49 ff 9d d8   b0
08e9 : 24 ca 10 f2 a2 1f bd 18   1f
08f1 : 09 9d d8 20 49 ff 9d d8   c0
08f9 : 24 ca 10 f2 4c 8b 17 db   1a
0901 : 3c 66 7e 66 66 66 00 c3   fe
0909 : 3c 66 66 66 66 3c 00 66   f4
0911 : 00 66 66 66 66 3c 00 66   c0
0919 : 00 3c 06 3e 66 3e 00 66   a6
0921 : 00 3c 66 66 66 3c 00 66   bb
0929 : 00 00 66 66 66 3e 00 3c   61
0931 : 66 66 6c 66 66 6c 60 a9   51
0939 : 88 a0 18 20 ae 14 a9 10   6e
0941 : a0 12 a2 0c 20 49 16 b0   1b
0949 : 2e f0 2c a9 96 a0 18 20   3f
0951 : ae 14 20 97 17 20 7d 09   7f
0959 : b0 e4 20 dc 14 20 f2 15   57
0961 : b0 dc a9 a7 a0 18 20 ae   87
0969 : 14 20 97 17 20 01 0a 20   c9
0971 : dc 14 a2 03 20 de 14 a2   ef
0979 : 0c 4c de 14 a2 00 bd 00   07
0981 : 02 9d 50 19 e8 ec 2e 17   66
0989 : d0 f4 a9 10 a2 50 a0 19   a1
0991 : 20 bd ff a0 00 a9 08 20   51
0999 : 86 17 20 c0 ff a2 08 20   40
09a1 : c6 ff 20 a5 ff aa a4 90   2d
09a9 : d0 29 8a d0 3c 20 a5 ff   26
09b1 : c9 08 d0 35 a9 e0 85 50   b1
09b9 : a9 76 85 51 20 a5 ff 20   98
09c1 : 91 17 a0 00 91 50 20 8b   39
09c9 : 17 e6 50 d0 02 e6 51 a6   6b
09d1 : 90 f0 e9 20 cc ff a9 08   db
09d9 : 20 c3 ff 20 91 17 a0 00   33
09e1 : a9 ff 91 50 20 8b 17 18   e3
09e9 : 60 20 cc ff a9 08 20 c3   6f
09f1 : ff a9 be a0 18 20 ae 14   ee
09f9 : 20 2e 15 20 dc 14 38 60   8a
0a01 : 20 91 17 a9 00 85 52 8d   75
0a09 : 62 19 a9 28 85 53 8d 63   57
0a11 : 19 a9 ff 8d 00 28 a9 e0   5a
0a19 : 85 50 a9 76 85 51 a9 00   89
0a21 : 8d 6a 19 a9 00 8d 6b 19   ab
0a29 : a9 00 8d 6c 19 a9 00 8d   bd
0a31 : 6d 19 20 68 0b b0 1a ee   bc
0a39 : 6a 19 d0 03 ee 6b 19 20   b3
0a41 : c3 0b 90 ee ee 6c 19 d0   e4
0a49 : 03 ee 6d 19 20 a1 0c 90   a2
0a51 : e1 20 dc 14 4c 8b 17 20   ba
0a59 : 91 17 a0 00 b1 50 e6 50   78
0a61 : d0 02 e6 51 4c 8b 17 ff   94
0a69 : 01 02 03 04 05 06 07 08   59
0a71 : 09 0a 0b 0c 0d 0e 0f 10   61
0a79 : 11 12 13 14 15 16 17 18   69
0a81 : 19 1a 1b 1c 1d 1e 00 00   b5
0a89 : 00 00 00 00 00 00 00 00   8a
0a91 : 00 00 00 00 00 00 00 00   92
0a99 : 00 00 00 00 00 00 00 00   9a
0aa1 : 00 00 00 00 00 00 00 00   a2
0aa9 : 01 02 03 04 05 06 07 08   99
0ab1 : 09 0a 0b 0c 0d 0e 0f 10   a1
0ab9 : 11 12 13 14 15 16 17 18   a9
0ac1 : 19 1a 1b 1c 1d 00 00 00   04
0ac9 : 00 00 00 00 00 00 00 00   ca
0ad1 : 00 00 00 00 00 00 00 00   d2
0ad9 : 00 00 00 00 00 00 00 00   da
0ae1 : 00 00 00 00 00 00 7f 00   df
0ae9 : 00 00 00 00 00 00 00 00   ea
0af1 : 00 00 00 00 00 00 00 00   f2
0af9 : 00 00 00 00 00 00 00 00   fa
0b01 : 00 00 00 00 00 00 00 00   02
0b09 : 00 00 00 00 00 00 00 00   0a
0b11 : 00 00 00 00 00 00 00 00   12
0b19 : 00 00 00 00 00 00 00 00   1a
0b21 : 00 00 00 00 00 00 00 00   22
0b29 : 00 00 00 00 00 00 00 00   2a
0b31 : 00 00 00 00 00 00 00 00   32
0b39 : 00 00 00 00 00 00 00 00   3a
0b41 : 00 00 00 00 00 00 00 00   42
0b49 : 00 00 00 00 00 00 00 00   4a
0b51 : 00 00 00 00 00 00 00 00   52
0b59 : 00 00 00 00 00 00 00 00   5a
0b61 : 00 00 00 00 00 00 ff a0   a3
0b69 : 00 b1 50 e6 50 d0 02 e6   94
0b71 : 51 a8 b9 68 0a f0 f0 30   de
0b79 : 47 a2 00 9d 00 1e e8 a0   9b
0b81 : 00 b1 50 e6 50 d0 02 e6   ac
0b89 : 51 a8 b9 68 0a c9 7f f0   79
0b91 : ee c9 ff f0 2b c9 00 d0   24
0b99 : e2 e0 02 90 ca 8e 6e 19   8b
0ba1 : a2 00 bd 00 1e 9d 78 04   6b
0ba9 : e8 e0 28 f0 11 ec 6e 19   8e
0bb1 : d0 f0 a9 20 e0 28 f0 06   87
0bb9 : 9d 78 04 e8 d0 f6 18 60   96
0bc1 : 38 60 a9 00 85 52 a9 28   76
0bc9 : 85 53 20 14 0c 20 5d 0d   d4
0bd1 : a5 52 8d 64 19 a5 53 8d   b6
0bd9 : 65 19 20 36 0c b0 11 20   65
0be1 : 6e 0c f0 0e b0 ea ad 64   35
0be9 : 19 85 52 ad 65 19 85 53   eb
0bf1 : 38 60 ee 60 19 d0 03 ee   23
0bf9 : 61 19 ad 64 19 85 52 ad   41
0c01 : 65 19 85 53 a0 02 ad 60   50
0c09 : 19 91 52 c8 ad 61 19 91   06
0c11 : 52 18 60 ac 6e 19 b9 ff   b4
0c19 : 1d aa bd c8 19 99 7f 1e   ac
0c21 : 88 d0 f3 60 ac 6f 19 b9   38
0c29 : ff 1e aa bd c8 19 99 7f   54
0c31 : 1f 88 d0 f3 60 a0 00 b1   b5
0c39 : 52 8d c1 19 30 2d c8 b1   d8
0c41 : 52 8d 6f 19 c8 b1 52 8d   d7
0c49 : 60 19 c8 b1 52 8d 61 19   e7
0c51 : c8 a2 00 b1 52 9d 00 1f   f1
0c59 : e8 c8 ec 6f 19 d0 f4 18   ea
0c61 : 98 65 52 85 52 90 02 e6   71
0c69 : 53 18 60 38 60 20 25 0c   9b
0c71 : ad 6e 19 cd 6f 19 90 03   5d
0c79 : ad 6f 19 8d 66 19 a2 00   90
0c81 : bd 80 1e dd 80 1f d0 13   2c
0c89 : e8 ec 66 19 d0 f2 ad 6e   dc
0c91 : 19 cd 6f 19 f0 05 b0 04   92
0c99 : 18 a9 01 60 38 a9 01 60   68
0ca1 : 18 ad 6e 19 69 04 18 6d   41
0ca9 : 62 19 8d 67 19 ad 63 19   a7
0cb1 : 69 00 8d 68 19 38 ad 67   64
0cb9 : 19 e9 de ad 68 19 e9 76   18
0cc1 : b0 28 20 ed 0c a0 00 98   42
0cc9 : 91 52 c8 ad 6e 19 91 52   06
0cd1 : 98 c8 91 52 c8 a9 00 91   79
0cd9 : 52 c8 a2 00 bd 00 1e 91   af
0ce1 : 52 e8 c8 ec 6e 19 d0 f4   54
0ce9 : 18 60 38 60 ad 62 19 85   a9
0cf1 : 58 ad 63 19 85 59 ad 67   c4
0cf9 : 19 85 5a ad 68 19 85 5b   3d
0d01 : e6 58 d0 02 e6 59 e6 5a   11
0d09 : d0 02 e6 5b 38 a5 58 e5   dd
0d11 : 52 a8 a5 59 e5 53 aa 8c   09
0d19 : 4e 19 38 a5 58 ed 4e 19   17
0d21 : 85 58 b0 02 c6 59 38 a5   a2
0d29 : 5a ed 4e 19 85 5a b0 02   23
0d31 : c6 5b 98 f0 08 88 b1 58   25
0d39 : 91 5a 98 d0 f8 8a f0 0f   fd
0d41 : c6 59 c6 5b 88 b1 58 91   6b
0d49 : 5a 98 d0 f8 ca d0 f1 ad   99
0d51 : 67 19 8d 62 19 ad 68 19   c7
0d59 : 8d 63 19 60 a0 00 b1 52   60
0d61 : 30 2f c8 b1 52 aa a0 04   96
0d69 : b1 52 a8 b9 c8 19 cd 80   32
0d71 : 1e f0 11 b0 1c 18 8a 69   e1
0d79 : 04 18 65 52 85 52 90 dc   14
0d81 : e6 53 d0 d8 a0 05 b1 52   fe
0d89 : a8 b9 c8 19 cd 81 1e 90   e6
0d91 : e4 60 a2 7f a9 00 9d 80   50
0d99 : 1d ca 10 fa 60 a2 00 bd   15
0da1 : 00 1e 9d 80 1d e8 ec 6e   d1
0da9 : 19 d0 f4 8e c4 19 60 a2   15
0db1 : 00 bd 00 1f 9d 80 1d e8   98
0db9 : ec 6f 19 d0 f4 8e c4 19   c6
0dc1 : 60 a2 08 20 c9 ff a2 00   a0
0dc9 : bd 00 1f dd 80 1d d0 0b   54
0dd1 : e8 ec c4 19 f0 05 ec 6f   4d
0dd9 : 19 d0 ed 8a 8e c5 19 20   e3
0de1 : a8 ff ad 6f 19 20 a8 ff   17
0de9 : ae c5 19 ec 6f 19 f0 0c   f9
0df1 : bd 00 1f 20 a8 ff ee c5   4c
0df9 : 19 4c e9 0d 20 b0 0d 4c   a9
0e01 : cc ff a9 00 85 52 a9 28   19
0e09 : 85 53 a2 01 8e c0 19 ca   e9
0e11 : 8e c6 19 20 5a 0e b0 0a   3a
0e19 : 20 27 15 c9 03 f0 06 4c   b4
0e21 : 14 0e 20 27 15 4c 29 0e   9e
0e29 : a2 06 20 de 14 e8 e0 17   ec
0e31 : d0 f8 60 a9 d7 a0 18 20   ee
0e39 : ae 14 ad 6b 19 ae 6a 19   ad
0e41 : 20 cd bd a9 f1 a0 18 20   b1
0e49 : ae 14 ad 6d 19 ae 6c 19   05
0e51 : 20 cd bd 20 27 15 4c 29   6a
0e59 : 0e 20 29 0e a9 06 8d 69   57
0e61 : 19 a9 00 8d 70 19 a9 13   9d
0e69 : 8d 71 19 20 36 0c b0 72   65
0e71 : ad c6 19 f0 05 ad c1 19   dd
0e79 : f0 f1 ac 70 19 ae 69 19   7a
0e81 : 20 43 16 ad 61 19 ae 60   d8
0e89 : 19 20 cd bd 18 ad 70 19   c0
0e91 : 69 05 a8 a2 00 bd 00 1f   27
0e99 : 91 d1 e8 c8 cc 71 19 f0   05
0ea1 : 05 ec 6f 19 d0 ef ad c0   e0
0ea9 : 19 f0 12 ad c1 19 f0 0d   37
0eb1 : 18 ad 70 19 69 05 a8 ae   9e
0eb9 : 69 19 20 37 11 ee 69 19   fe
0ec1 : ad 69 19 c9 17 d0 a4 ad   88
0ec9 : 70 19 c9 14 f0 12 a9 06   0d
0ed1 : 8d 69 19 a9 14 8d 70 19   30
0ed9 : a9 27 8d 71 19 4c 6c 0e   69
0ee1 : 18 60 38 60 a2 00 dd f7   d5
0ee9 : 0e f0 07 e8 bc f7 0e d0   b4
0ef1 : f5 60 bd 07 0f 60 5a da   7a
0ef9 : 59 d9 2f 3c 3e 3a 3b 40   b5
0f01 : 5c 2d 2b 27 23 00 59 d9   ef
0f09 : 5a da 2d 3b 3a 5c 5b 5d   32
0f11 : 27 23 5e 2f 40 a9 00 84   b4
0f19 : c6 19 20 f9 10 20 19 11   3c
0f21 : 20 dd 10 b0 1b 20 63 0f   a8
0f29 : b0 16 ad 00 1e 30 1a 20   5c
0f31 : 3f 12 b0 15 20 9e 0d 20   b4
0f39 : 6e 0c 90 ee d0 0b f0 e0   9a
0f41 : 20 aa 0f 20 29 0e 4c dc   6c
0f49 : 14 20 31 11 20 fd 12 20   56
0f51 : dd 10 b0 ec 20 63 0f b0   bb
0f59 : e7 ad 00 1e 30 eb f0 ca   96
0f61 : 10 d5 ad 74 19 f0 13 ad   16
0f69 : 00 1f 20 df 11 cd 75 19   84
0f71 : f0 35 20 aa 0f a9 00 8d   b3
0f79 : 00 1e a9 01 8d 74 19 20   34
0f81 : 93 0d ad 00 1f 20 df 11   9b
0f89 : 8d 75 19 8d 81 19 a9 0c   68
0f91 : a2 76 a0 19 20 bd ff a0   eb
0f99 : 02 a9 01 20 86 17 20 c0   d7
0fa1 : ff 20 f2 15 90 01 60 18   d2
0fa9 : 60 20 cc ff a9 01 4c c3   a8
0fb1 : ff a9 ff 4c 49 10 a9 00   ca
0fb9 : 85 52 a9 28 85 53 a9 01   72
0fc1 : 8d c0 19 a9 ff 8d c6 19   e4
0fc9 : 4c 14 0e a9 01 a2 c7 a0   5e
0fd1 : 19 20 bd ff a9 04 a2 04   b7
0fd9 : a0 07 20 ba ff 20 c0 ff   60
0fe1 : b0 36 a2 04 20 c9 ff a6   73
0fe9 : 90 d0 2d a9 00 85 52 a9   2b
0ff1 : 28 85 53 20 36 0c b0 20   7c
0ff9 : ad c1 19 f0 f6 a2 00 bd   eb
1001 : 00 1f 20 21 10 e8 ec 6f   98
1009 : 19 d0 f4 a9 0d 20 d2 ff   1a
1011 : a9 00 20 d2 ff 4c f4 0f   71
1019 : 20 cc ff a9 04 4c c3 ff   86
1021 : a8 b9 28 10 4c d2 ff 40   8e
1029 : 41 42 43 44 45 46 47 48   19
1031 : 49 4a 4b 4c 4d 4e 4f 50   21
1039 : 51 52 53 54 55 56 57 58   29
1041 : 59 5a 7b 7c 7d 7e a9 00   a8
1049 : 8d c6 19 20 f9 10 20 19   57
1051 : 11 20 dd 10 b0 1c 20 72   3d
1059 : 11 b0 1f ad 00 1e 30 26   3e
1061 : 20 3f 12 b0 21 20 6e 0c   a0
1069 : 90 59 d0 1a 20 96 12 4c   b5
1071 : 52 10 20 e4 12 a2 03 20   f2
1079 : de 14 20 f9 11 20 15 12   33
1081 : 20 29 0e 4c dc 14 a9 0b   6e
1089 : a0 19 20 ae 14 20 ca 10   21
1091 : 20 2e 15 c9 03 f0 db c9   02
1099 : 4e f0 22 c9 4a d0 06 20   a5
10a1 : 2a 13 20 c2 0d a2 03 20   e7
10a9 : de 14 20 dd 10 b0 c3 20   2b
10b1 : 72 11 b0 c6 ad 00 1e 30   64
10b9 : cd f0 a0 10 a8 20 fd 12   d0
10c1 : 4c a6 10 20 96 12 4c 5c   4c
10c9 : 10 a2 00 bd 00 1f 9d 78   42
10d1 : 04 e8 e0 28 f0 05 ec 6f   50
10d9 : 19 d0 f0 60 20 31 11 20   b3
10e1 : 4e 11 20 36 0c b0 0b ad   55
10e9 : c6 19 f0 05 ad c1 19 f0   48
10f1 : f1 18 08 20 31 11 28 60   f1
10f9 : a9 00 85 52 a9 28 85 53   e7
1101 : a9 05 8d 72 19 a9 05 8d   ed
```

```
1109 : 73 19 a9 00 8d 74 19 8d  6f
1111 : c0 19 8d 00 1e 4c 31 11  ec
1119 : a5 52 8d 64 19 a5 53 8d  fe
1121 : 65 19 20 5a 0e ad 64 19  78
1129 : 85 52 ad 65 19 85 53 60  bb
1131 : ae 72 19 ac 73 19 bd f5  d7
1139 : 14 85 d1 bd 0e 15 85 d2  81
1141 : a2 0d b1 d1 49 80 91 d1  93
1149 : c8 ca 10 f6 60 ee 72 19  d3
1151 : ad 72 19 c9 17 d0 11 a9  46
1159 : 06 8d 72 19 ad 73 19 c9  54
1161 : 19 f0 06 a9 19 8d 73 19  a7
1169 : 60 a9 05 8d 73 19 4c 19  f4
1171 : 11 ad 74 19 f0 1b ad 00  38
1179 : 1f 20 df 11 cd 75 19 f0  91
1181 : 5b 20 e4 12 20 f9 11 20  be
1189 : 15 12 b0 4f a9 00 8d 00  8e
1191 : 1e a9 01 8d 74 19 20 93  2e
1199 : 0d ad 00 1f 20 df 11 8d  c1
11a1 : 75 19 8d 81 19 8d ae 19  21
11a9 : 20 1e 12 b0 2e a9 0c a2  19
11b1 : 76 a0 19 20 bd ff a0 01  22
11b9 : a9 08 20 86 17 20 c0 ff  b5
11c1 : 20 f2 15 b0 16 a9 0f a2  e6
11c9 : 82 a0 19 20 bd ff a0 02  48
11d1 : a9 01 20 86 17 20 c0 ff  49
11d9 : 4c f2 15 60 18 60 a2 03  05
11e1 : dd f1 11 f0 05 ca 10 f8  f2
11e9 : 30 03 bd f5 11 09 40 60  e4
11f1 : 1b 1c 1d 1e 01 0f 15 13  29
11f9 : a2 08 20 c9 ff a9 ff 20  6e
1201 : a8 ff 20 cc ff a9 08 20  f8
1209 : c3 ff a9 01 20 c3 ff a9  ca
1211 : 0f 4c c3 ff a9 11 a2 af  44
1219 : a0 19 4c 24 12 a9 1e a2  0a
1221 : 91 a0 19 20 bd ff a9 0f  ed
1229 : a2 08 a8 20 ba ff a9 0f  6e
1231 : 20 c3 ff 20 c0 ff a9 0f  07
1239 : 20 c3 ff 4c f2 15 a2 01  29
1241 : 20 c6 ff a2 00 8e c5 19  d6
1249 : 8e c2 19 20 a5 ff 30 38  0e
1251 : f0 14 8d c4 19 a2 00 bd  69
1259 : 80 1d 9d 00 1e e8 ec c4  36
1261 : 19 d0 f4 8e c5 19 20 a5  e2
1269 : ff 8d 6e 19 ae c5 19 ec  45
1271 : 6e 19 f0 0c 20 a5 ff 9d  94
1279 : 00 1e ee c5 19 4c 6d 12  ca
1281 : 20 cc ff 20 14 0c 18 60  ce
1289 : 8d 00 1e 20 cc ff a2 05  03
1291 : 20 de 14 38 60 a9 01 8d  9f
1299 : c2 19 ad 00 1e 30 43 f0  a6
12a1 : 41 a2 08 20 c9 ff a2 00  61
12a9 : bd 00 1e dd 80 1d d0 0b  f4
12b1 : e8 ec c4 19 f0 05 ec 6e  2b
12b9 : 19 d0 ed 8a 8e c5 19 20  c3
12c1 : a8 ff ad 6e 19 20 a8 ff  d7
12c9 : ae c5 19 ec 6e 19 f0 0c  c9
12d1 : bd 00 1e 20 a8 ff ee c5  ec
12d9 : 19 4c c9 12 20 cc ff 4c  ce
12e1 : 9e 0d 60 ad 00 1e 30 13  ab
12e9 : ad c2 19 d0 03 20 96 12  07
12f1 : 20 3f 12 b0 06 20 96 12  2b
12f9 : 4c f1 12 60 38 a5 52 e9  9c
1301 : 04 85 52 b0 02 c6 53 38  87
1309 : a5 52 ed 6f 19 85 52 b0  a9
1311 : 02 c6 53 a0 00 a9 01 91  d4
1319 : 52 18 ad 6f 19 69 04 18  ee
1321 : 65 52 85 52 90 02 e6 53  b6
1329 : 60 38 a5 52 e9 04 85 52  d2
1331 : b0 02 c6 53 38 a5 52 ed  d4
1339 : 6f 19 85 52 b0 02 c6 53  bd
1341 : a0 00 98 91 52 18 ad 6f  b5
1349 : 19 69 04 18 65 52 85 52  bf
1351 : 90 02 e6 53 60 a9 27 a0  38
1359 : 00 a2 03 20 49 16 b0 55  22
1361 : f0 53 ae 2e 17 bd 00 02  d0
1369 : 29 1f 9d 00 1f ca 10 f5  fe
1371 : 20 93 0d ad 00 1f 20 df  8d
1379 : 11 8d 81 19 a9 0c a2 76  47
1381 : a0 19 20 bd ff a0 02 a9  ce
1389 : 01 20 86 17 20 c0 ff 20  67
1391 : f2 15 b0 21 20 c3 13 20  0b
1399 : 3f 12 b0 10 20 9e 0d 20  7b
13a1 : c9 13 d0 f3 20 e7 13 b0  95
13a9 : 0c 4c 98 13 20 27 15 20  34
13b1 : b6 13 4c 56 13 a9 01 20  91
13b9 : c3 ff 20 29 0e a2 03 4c  44
13c1 : de 14 a9 06 8d 69 19 60  1e
13c9 : ad 6e 19 cd 2e 17 90 13  b1
13d1 : a2 00 bd 00 1e dd 00 1f  f2
13d9 : d0 09 e8 ec 2e 17 d0 f2  ca
13e1 : a9 00 60 a9 01 60 ae 69  78
13e9 : 19 bd f5 14 85 d1 bd 0e  db
13f1 : 15 85 d2 a0 00 a2 00 bd  22
13f9 : 00 1e 91 d1 c8 e8 c0 28  ce
1401 : f0 05 ec 6e 19 d0 f0 ee  36
1409 : 69 19 ad 69 19 c9 17 d0  75
1411 : 0f a9 06 8d 69 19 20 27  56
1419 : 15 c9 03 f0 05 20 29 0e  04
1421 : 18 60 38 60 a9 27 a0 00  da
1429 : a2 03 20 49 16 b0 2d f0  fc
1431 : 2b ae 2e 17 bd 00 02 29  58
1439 : 1f 9d 00 1f ca 10 f5 a9  63
1441 : 00 8d 74 19 20 72 11 20  62
1449 : 3f 12 b0 10 20 c9 13 f0  3e
1451 : 06 20 96 12 4c 48 14 20  e7
1459 : 68 14 90 eb 20 f9 11 a2  c8
1461 : 03 20 de 14 4c 15 12 a2  a9
1469 : 03 20 de 14 a2 00 bd 00  d8
1471 : 1e 9d 78 04 e8 e0 28 f0  15
1479 : 05 ec 6e 19 d0 f0 a9 31  51
1481 : a0 19 20 ae 14 20 2e 15  b1
1489 : c9 4a f0 07 c9 03 f0 0d  27
1491 : 20 96 12 a2 03 20 de 14  aa
1499 : 20 9e 0d 18 60 20 96 12  d4
14a1 : a2 03 20 de 14 20 9e 0d  80
14a9 : 20 e4 12 38 60 8d d1 14  a9
14b1 : 8c d2 14 20 d0 14 aa e0  ca
14b9 : 0d f0 1f 20 d0 14 a8 20  9b
14c1 : 43 16 20 d0 14 c9 40 f0  a4
14c9 : ea 20 d2 ff 4c c3 14 ad  06
14d1 : ff ff ee d1 14 d0 03 ee  77
14d9 : d2 14 60 a2 18 bd f5 14  91
14e1 : 8d ef 14 bd 0e 15 8d f0  c4
14e9 : 14 a0 27 a9 20 99 ff ff  1b
14f1 : 88 10 fa 60 00 28 50 78  bf
14f9 : a0 c8 f0 18 40 68 90 b8  37
1501 : e0 08 30 58 80 a8 d0 f8  7f
1509 : 20 48 70 98 c0 04 04 04  c1
1511 : 04 04 04 04 05 05 05 05  30
1519 : 05 05 06 06 06 06 06 06  98
1521 : 06 07 07 07 07 07 a9 6c  76
1529 : a0 18 20 ae 14 a9 00 85  4d
1531 : c6 20 a0 17 20 e4 ff f0  1d
1539 : fb 48 20 dc 14 68 60 a9  55
1541 : 00 85 c6 a5 c6 f0 fc 60  13
1549 : a9 01 a2 4d a0 19 20 bd  94
1551 : ff a0 00 a9 08 20 86 17  9f
1559 : 20 c0 ff a2 08 20 c6 ff  ca
1561 : 20 bc 15 20 bc 15 a9 06  50
1569 : d0 02 a9 07 8d 4f 19 20  7e
1571 : c9 15 ae 4f 19 a0 07 20  4d
1579 : 43 16 20 bc 15 20 bc 15  d6
1581 : 20 bc 15 8d 4e 19 20 bc  9e
1589 : 15 ae 4e 19 20 cd bd 20  54
1591 : bc 15 20 d2 ff d0 f8 a9  f8
1599 : 0d 20 d2 ff ee 4f 19 ad  94
15a1 : 4f 19 c9 17 d0 cc 20 e0  88
15a9 : 15 a6 90 f0 bd d0 03 20  02
15b1 : e0 15 20 d8 15 20 f2 15  87
15b9 : 4c 29 0e 20 a5 ff a6 90  38
15c1 : f0 05 68 68 4c b0 15 60  ba
15c9 : a2 07 20 de 14 e8 e0 17  0d
15d1 : d0 f8 60 04 04 00 ff 20  36
15d9 : cc ff a9 08 4c c3 ff 20  33
15e1 : cc ff 20 27 15 c9 03 d0  e7
15e9 : 03 4c b3 15 a2 08 4c c6  cb
15f1 : ff a9 0f a2 08 a8 20 ba  99
15f9 : ff a9 00 20 bd ff 20 c0  af
1601 : ff a2 0f 20 c6 ff a0 00  08
1609 : 20 a5 ff a6 90 d0 06 99  ab
1611 : 00 02 c8 d0 f3 a9 00 99  1e
1619 : 00 02 20 cc ff ad 00 02  2d
1621 : 0d 01 02 c9 30 d0 02 18  2a
1629 : 60 a0 00 a2 18 20 43 16  e9
1631 : a2 00 bd 00 02 20 d2 ff  af
1639 : f0 03 e8 d0 f5 20 2e 15  42
1641 : 38 60 18 4c f0 ff 38 60  ea
1649 : 8d 2d 17 8c 2b 17 84 d3  e9
1651 : bd f5 14 85 d1 bd 0e 15  2c
1659 : 85 d2 18 ad 2b 17 6d 2d  7f
1661 : 17 8d 2c 17 85 d5 a9 00  da
1669 : 85 c6 a5 c6 85 cc 8d 92  ae
1671 : 02 f0 f7 78 a5 cf f0 0e  b1
1679 : a5 ce ae 87 02 a0 00 84  50
1681 : d4 84 cf 20 13 ea 20 b4  02
1689 : e5 20 e5 0e c9 03 f0 b6  9f
1691 : c9 14 d0 0a a6 d3 ec 2b  ed
1699 : 17 f0 cf 4c 38 17 c9 0d  23
16a1 : f0 47 c9 1d d0 0b a6 d3  f2
16a9 : ec 2c 17 f0 bd e6 d3 d0  93
16b1 : 2c c9 20 90 b5 c9 9d d0  9e
16b9 : 09 a6 d3 ec 2b 17 f0 aa  2c
16c1 : d0 18 c9 94 d0 03 4c 65  c3
16c9 : 17 c9 80 90 06 c9 a0 b0  8a
16d1 : 02 90 97 a6 d3 ec 2c 17  59
16d9 : b0 90 20 16 e7 a5 d3 cd  33
16e1 : 2c 17 d0 02 c6 d3 4c 6b  20
16e9 : 16 20 2f 17 ac 2c 17 b1  aa
16f1 : d1 c9 20 d0 0c 88 cc 2b  58
16f9 : 17 d0 f4 b1 d1 c9 20 f0  b9
1701 : 1f c8 8c 2e 17 a2 00 ac  4d
1709 : 2b 17 b1 d1 20 50 17 9d  82
1711 : 00 02 e8 c8 cc 2e 17 d0  a2
1719 : f1 ad 00 02 c9 20 d0 02  06
1721 : a2 00 8e 2e 17 ad 2e 17  f2
1729 : 18 60 00 00 00 00 a4 d3  ac
1731 : b1 d1 29 7f 91 d1 60 a4  78
1739 : d3 c6 d3 b1 d1 88 91 d1  e6
1741 : c8 c8 cc 2c 17 d0 f4 88  03
1749 : a9 20 91 d1 4c 6b 16 c9  ad
1751 : 20 90 09 c9 40 90 04 c9  61
1759 : 60 90 05 60 18 69 40 60  dd
1761 : 18 69 80 60 ac 2c 17 88  f4
1769 : b1 d1 c9 20 d0 14 c4 d3  e2
1771 : f0 10 c8 88 88 b1 d1 c8  9b
1779 : 91 d1 c4 d3 d0 f5 a9 20  42
1781 : 91 d1 4c 6b 16 a2 08 4c  ab
1789 : ba ff a0 37 84 01 58 60  c4
1791 : 78 a0 30 84 01 60 a9 11  d2
1799 : a2 09 a0 80 4c a6 17 a9  a2
17a1 : 15 a2 19 a0 fc 8d 01 d4  4c
17a9 : 8e 05 d4 8c 06 d4 a9 11  50
17b1 : 8d 00 d4 a9 11 8d 04 d4  e0
17b9 : a9 0f 8d 18 d4 a2 a0 a0  76
17c1 : 00 88 d0 fd ca d0 f8 a9  63
17c9 : 10 8d 04 d4 60 00 00 93  69
17d1 : 12 cd 41 53 54 45 52 2d  98
17d9 : d3 50 45 4c 4c 20 56 32  33
17e1 : 2e 31 20 20 20 20 20 20  78
17e9 : 20 20 42 59 20 c4 49 45  ad
17f1 : 54 45 52 20 c2 41 59 45  a7
17f9 : 52 92 40 01 01 43 20 44  f8
1801 : 20 45 20 46 20 49 20 4c  fa
1809 : 20 55 20 57 20 c3 20 cd  03
1811 : 20 d0 20 40 02 00 c0 c0  4e
1819 : c0 c0 c0 c0 c0 c0 c0 c0  18
1821 : c0 c0 c0 c0 c0 c0 c0 c0  20
1829 : c0 c0 c0 c0 c0 c0 c0 c0  28
1831 : c0 c0 c0 c0 c0 c0 c0 c0  30
1839 : c0 c0 c0 c0 c0 c0 40 04  bd
1841 : 00 c0 c0 c0 c0 c0 c0 c0  80
1849 : c0 c0 c0 c0 c0 c0 c0 c0  48
1851 : c0 c0 c0 c0 c0 c0 c0 c0  50
1859 : c0 c0 c0 c0 c0 c0 c0 c0  58
1861 : c0 c0 c0 c0 c0 c0 c0 c0  60
1869 : c0 40 0d 18 08 c2 49 54  f4
1871 : 54 45 20 45 49 4e 45 20  75
1879 : d4 41 53 54 45 20 44 52  58
1881 : 5d 43 4b 45 4e 40 0d 0c  2f
1889 : 07 d4 45 58 54 4e 41 4d  ae
1891 : 45 20 3a 40 0d 18 0d 4c  db
1899 : 41 44 45 20 44 45 4e 20  3a
18a1 : d4 45 58 54 40 0d 18 0a  99
18a9 : 47 45 4e 45 52 49 45 52  f8
18b1 : 45 20 d7 4f 52 54 4c 49  72
18b9 : 53 54 45 40 0d 18 09 cb  dd
18c1 : 45 49 4e 20 cd 41 53 54  1f
18c9 : 45 52 2d d4 45 58 54 2d  e0
18d1 : c6 49 4c 45 40 0d 0c 05  9e
18d9 : c1 4e 5a 41 48 4c 20 41  6a
18e1 : 4c 4c 45 52 20 d7 5c 52  c6
18e9 : 54 45 52 20 3a 20 40 0d  38
18f1 : 0e 05 d6 45 52 53 43 48  3d
18f9 : 49 45 44 45 4e 45 20 d7  de
1901 : 5c 52 54 45 52 20 3a 20  93
1909 : 40 0d 18 03 d7 4f 52 54  20
1911 : 20 55 4e 42 45 4b 41 4e  08
1919 : 4e 54 21 20 cc 45 52 4e  bb
1921 : 45 4e 20 3f 20 28 4a 41  6c
1929 : 2f 4e 45 49 4e 29 40 0d  43
1931 : 18 08 d7 4f 52 54 20 4c  0e
1939 : 5c 53 43 48 45 4e 20 3f  de
1941 : 20 28 4a 41 2f 4e 45 49  3d
1949 : 4e 29 40 0d 24 00 00 31  82
1951 : 32 33 34 35 36 37 38 39  41
1959 : 30 31 32 33 34 35 54 00  53
1961 : 00 00 00 00 00 00 00 00  62
1969 : 00 00 00 00 00 00 00 00  6a
1971 : 00 00 00 00 00 53 50 45  d8
1979 : 4c 4c 2e 44 49 43 54 2e  5c
1981 : 41 53 50 45 4c 4c 2e 44  91
1989 : 49 43 54 2e 48 45 4c 50  cf
1991 : 52 3a 53 50 45 4c 4c 2e  23
1999 : 44 49 43 54 2e 48 45 4c  b0
19a1 : 50 3d 53 50 45 4c 4c 2e  b3
19a9 : 44 49 43 54 2e 41 53 3a  9c
19b1 : 53 50 45 4c 4c 2e 44 49  e1
19b9 : 43 54 2e 48 45 4c 50 00  b3
19c1 : 00 00 00 00 00 00 00 00  c2
19c9 : 01 03 04 05 06 07 08 09  b8
19d1 : 0a 0b 0c 0d 0e 0f 10 12  c3
19d9 : 13 14 15 17 18 1a 1b 1c  16
19e1 : 1d 1e 02 11 19 16 ff 00  f2
```

Listing 1. »MASTER-SPELL« bereitet Tippfehlern ein Ende

Master-Text und MPS 801/803: verbesserte Druckroutine

Das Textverarbeitungsprogramm »Master-Text« kann zwar reverse Schrift generieren, aber nicht für die Drucker MPS 801/803. Eine neue Druckroutine setzt diesem Manko ein Ende.

Die in Ausgabe 2/87 vorgestellte Druckeranpassung für Master-Text und den Commodore-Drucker MPS 803 erlaubte keinen Ausdruck in Revers-Schrift; diese Routine kann auch das. Man hat damit alle Vorteile der bisherigen Routine, also Drucken in Breitschrift, Unterstreichen und das Anführungszeichen unten und oben. Zusätzlich erhält man die Möglichkeit der Reversschrift.

Hier noch ein paar Besonderheiten: Bei der Breitschrift muß man ein wenig aufpassen, da die doppelte Breite der Zeichen vom Hauptprogramm nicht erkannt wird. Dadurch kann es Probleme geben, wenn die Zeile zu lang wird. Steuerzeichen können auch innerhalb der Anführungszeichen benutzt werden, weil dafür ein anderer Code zum Drucker geschickt wird. Leerzeichen am Anfang einer Zeile werden normalerweise nicht unterstrichen oder revers gedruckt. Wer dies dennoch möchte, muß anstatt ⟨SPACE⟩ ein ⟨SHIFT SPACE⟩ setzen. Das Programm (Listing 1) belegt den Speicher von $9A00-$A000 und zusätzlich noch den Speicher von $C400-$C600 zum Zwischenspeichern von Druckerdaten.

Nun zur eigentlichen Bedienung, die sehr einfach ist: Zunächst muß die neue Druckerroutine abgetippt und unter dem üblichen Namen »UMLAUT1« gespeichert werden. Sie wird dann wie gewohnt geladen, es braucht also an der Laderoutine nichts geändert zu werden. Auch nicht, wenn mit dem »INST-PACKER« gearbeitet wird. Es müssen dann nur noch einige Druckercodes geändert werden. Dazu wird Master-Text geladen und im Dienstmenü der Punkt »Drucker« gewählt. In der Steuerzeichentabelle müssen folgende Werte eingegeben werden:

Reversschrift	12 und 92,
Breitschrift	0e und 0f (wie gewohnt),
Unterstreichen	0b und 0c.

Im Menü »Druckertabelle« folgt die Eingabe der Codes für die Sonderzeichen:

ä = 166	Ü = 190	
Ä = 168	ß = 191	Sonderzeichen = 187
ö = 223	§ = 220	⟨SHIFT SPACE⟩ = 160
Ö = 172	„ = 126	
ü = 188		

Das Sonderzeichen für Code 187 kann mit dem Zeicheneditor definiert werden und z.B. auf die Tastenkombination ⟨CBM*⟩ gelegt werden. Zu beachten ist noch, daß die drei Schriftarten so lange eingeschaltet bleiben, bis sie wieder abgeschaltet werden. Daher ist es wichtig, am Schluß einer Seite das Abschalten vorzusehen, wenn mehrere oder verschiedene Ausdrucke gewünscht werden.

(Guido Barnert/Dr. Rudolf Egg)

```
Name : umlaut1               9a00 a000
--------------------------------------
9a00 : a9 19 8d 26 03 a9 9a 8d   61
9a08 : 27 03 a9 00 a2 00 9d 00   bc
9a10 : c4 9d 00 c5 e8 d0 f7 60   11
9a18 : 00 48 a5 9a c9 04 f0 04   81
9a20 : 68 4c ca f1 68 8e a1 c4   aa
9a28 : 8c a2 c4 8d a0 c4 c9 0d   59
9a30 : d0 03 4c 03 9b c9 0a f0   07
9a38 : f9 c9 22 d0 02 18 60 c9   ae
9a40 : 0e d0 05 8d a4 c4 18 60   3b
9a48 : c9 0f f0 f7 c9 0b d0 05   16
9a50 : 8d a6 c4 18 60 c9 0c d0   8b
9a58 : 09 48 a9 00 8d a6 c4 68   e2
9a60 : 18 60 c9 12 d0 05 8d ab   20
9a68 : c4 18 60 c9 92 d0 09 48   ee
9a70 : a9 00 8d ab c4 68 18 60   a3
9a78 : c9 20 f0 05 a0 01 8c ad   ce
9a80 : c4 18 c9 20 90 09 18 c9   0c
9a88 : a0 b0 06 c9 80 90 02 18   00
9a90 : 60 c9 7e d0 14 ad a5 c4   5d
9a98 : f0 0a a9 00 8d a5 c4 a9   64
9aa0 : 7e 4c a9 9a a9 a2 8d a5   33
9aa8 : c4 48 ad ad c4 f0 18 ad   41
9ab0 : a4 c4 c9 0e d0 11 48 ad   fd
9ab8 : a3 c4 29 01 f0 05 a9 20   46
9ac0 : 20 c8 9a 68 20 c8 9a 68   7b
9ac8 : 18 ae a3 c4 e0 50 90 0b   a2
9ad0 : 48 20 03 9b 68 ac a2 c4   5d
9ad8 : ae a3 c4 9d 00 c4 ad ad   75
9ae0 : c4 f0 12 ad a6 c4 f0 03   b1
9ae8 : 9d 50 c4 ad ab c4 18 7d   d0
9af0 : 50 c4 9d 50 c4 ee a3 c4   f0
9af8 : 18 ad a0 c4 ae a1 c4 ac   0c
9b00 : a2 c4 60 a9 00 8d a7 c4   e6
9b08 : 8d a8 c4 8d aa c4 8d ac   2c
9b10 : c4 a9 00 a2 0d 9d af c4   03
9b18 : ca d0 fa ae a7 c4 bd 00   76
9b20 : c5 c9 0e f0 57 a2 0a dd   da
9b28 : f5 9e f0 06 ca d0 f8 4c   19
9b30 : 4c 9b 8a 0a 0a 0a 38 e9   d3
9b38 : 08 8d 3f 9b a2 06 bd 30   fc
9b40 : 9f 9d b0 c4 ca d0 f7 a9   d9
9b48 : 01 8d a8 c4 ae a7 c4 bd   89

9b50 : 01 c5 a2 0a dd f5 9e f0   08
9b58 : 06 ca d0 f8 4c b3 9b 8a   fc
9b60 : 0a 0a 0a 38 e9 08 8d 6c   e7
9b68 : 9b a2 06 bd 48 9f 9d b6   f3
9b70 : c4 ca d0 f7 a9 01 8d a8   f6
9b78 : c4 4c b3 9b bd 01 c5 a2   03
9b80 : 0a dd f5 9e f0 06 ca d0   d6
9b88 : f8 4c b3 9b 8a 0a 0a 0a   3c
9b90 : 38 e9 08 8d 99 9b a2 06   7e
9b98 : bd 10 9f 48 8a 0a aa 68   c3
9ba0 : 9d af c4 9d b0 c4 48 8a   61
9ba8 : 4a aa 68 ca d0 ea a9 01   c8
9bb0 : 8d a8 c4 ae a7 c4 bd 50   d0
9bb8 : c5 f0 29 48 c9 12 f0 0d   54
9bc0 : a2 06 a9 02 1d b0 c4 9d   b6
9bc8 : b0 c4 ca d0 f5 a9 01 8d   73
9bd0 : a8 c4 68 c9 12 90 0d a2   4d
9bd8 : 06 a9 87 5d b0 c4 9d b0   49
9be0 : c4 ca d0 f5 ae a7 c4 bd   b3
9be8 : 51 c5 f0 29 48 c9 12 f0   7a
9bf0 : 0d a2 06 a9 02 1d b6 c4   73
9bf8 : 9d b6 c4 ca d0 f5 a9 01   e0
9c00 : 8d a8 c4 68 c9 12 90 0d   a9
9c08 : a2 06 a9 87 5d b6 c4 9d   e2
9c10 : b6 c4 ca d0 f5 ae a7 c4   f2
9c18 : bd 00 c4 c9 0e d0 03 4c   4c
9c20 : cb 9c 20 28 9c 4c 33 9c   79
9c28 : c9 ac d0 02 a9 a8 c9 be   40
9c30 : f0 fa 60 a2 03 dd 4f 9f   a5
9c38 : f0 06 ca d0 f8 4c 5d 9c   99
9c40 : 8a 0a 0a 0a 18 69 50 8d   bc
9c48 : 4d 9c a2 06 bd 68 9f 1d   25
9c50 : b0 c4 9d b0 c4 ca d0 f4   af
9c58 : a9 01 8d a8 c4 ae a7 c4   e4
9c60 : bd 50 c4 c9 12 90 12 a2   e3
9c68 : 06 a9 e0 5d b0 c4 9d b0   30
9c70 : c4 ca d0 f5 a9 01 8d a8   b6
9c78 : c4 ae a7 c4 bd 01 c4 20   4d
9c80 : 28 9c a2 03 dd 4f 9f f0   b8
9c88 : 06 ca d0 f8 4c ac 9c 8a   f8
9c90 : 0a 0a 0a 18 69 50 8d 9c   ad
9c98 : 9c a2 06 bd 58 9f 1d b6   23
9ca0 : c4 9d b6 c4 ca d0 f4 a9   d3
9ca8 : 01 8d a8 c4 ae a7 c4 bd   e9

9cb0 : 51 c4 c9 12 90 72 a2 06   4b
9cb8 : a9 e0 5d b6 c4 9d b6 c4   9d
9cc0 : ca d0 f5 a9 01 8d a8 c4   4e
9cc8 : 4c 28 9d bd 01 c4 20 28   4e
9cd0 : 9c a2 03 dd 4f 9f f0 06   fc
9cd8 : ca d0 f8 4c 0c 9d 8a 0a   be
9ce0 : 0a 0a 69 50 8d ea 9c a2   3c
9ce8 : 06 bd 58 9f 48 8a 0a aa   2d
9cf0 : 68 48 1d af c4 9d af c4   3b
9cf8 : 68 1d b0 c4 9d b0 c4 48   b6
9d00 : 8a 4a aa 68 ca d0 e2 a9   79
9d08 : 01 8d a8 c4 ae a7 c4 bd   49
9d10 : 51 c4 c9 12 90 12 a2 0c   b4
9d18 : a9 e0 5d b0 c4 9d b0 c4   24
9d20 : ca d0 f5 a9 01 8d a8 c4   ae
9d28 : ad a8 c4 d0 1b ee aa c4   d2
9d30 : ee a7 c4 ee a7 c4 ad a7   a7
9d38 : c4 c9 50 b0 03 4c 11 9b   19
9d40 : a9 08 20 ca f1 4c 78 9d   ed
9d48 : ad aa c4 f0 0f a9 0e 20   50
9d50 : ca f1 a9 20 20 ca f1 ce   3f
9d58 : aa c4 d0 f8 a9 08 20 ca   a8
9d60 : f1 a2 01 bd b0 c4 09 80   f1
9d68 : 20 ca f1 e8 e0 0d d0 f3   28
9d70 : a2 00 8e a8 c4 4c 30 9d   76
9d78 : a9 00 8d a7 c4 a9 0d 20   88
9d80 : ca f1 ad a9 c4 20 ca f1   40
9d88 : ad ac c4 20 ca f1 a9 11   c5
9d90 : 20 ca f1 ae a7 c4 bd 00   ff
9d98 : c4 c9 0e f0 02 a9 0f cd   28
9da0 : a9 c4 f0 06 8d a9 c4 20   22
9da8 : ca f1 c9 0e d0 04 ee a7   d7
9db0 : c4 e8 ea ea ea ea ea bd   2d
9db8 : 50 c4 c9 12 90 0a a9 12   43
9dc0 : cd ac c4 f0 10 4c cf 9d   10
9dc8 : a9 92 cd ac c4 f0 06 8d   ca
9dd0 : ac c4 20 ca f1 bd 00 c4   d6
9dd8 : a2 10 dd 6f 9f f0 4d ca   34
9de0 : d0 f8 c9 a8 d0 02 a9 c1   fb
9de8 : c9 ac d0 02 a9 cf c9 be   39
9df0 : d0 02 a9 d5 20 ca f1 ee   e4
9df8 : a7 c4 18 ad a7 c4 cd a3   dc
9e00 : c4 90 90 a9 08 20 ca f1   f6
9e08 : a9 00 8d a3 c4 8d ad c4   82
```

```
9e10 : 8d a7 c4 a2 a0 bd ff c3  76
9e18 : 9d ff c4 a9 00 9d ff c3  90
9e20 : ca d0 f2 18 a9 0d 20 ca  2b
9e28 : f1 4c f9 9a 8a 0a 0a 0a  46
9e30 : 18 69 78 8d 40 9e a9 08  7c
9e38 : 20 ca f1 a2 01 8a 48 bd  8f
9e40 : e0 9f ac ac c4 c0 12 d0  ed
9e48 : 02 49 7f 20 ca f1 ae a9  1d
9e50 : c4 e0 0e d0 03 20 ca f1  62
9e58 : 68 aa e8 e0 08 d0 de ad  49
9e60 : a9 c4 20 ca f1 ad ac c4  95
9e68 : 4c f4 9d 00 00 00 00 00  96
9e70 : 00 00 00 00 00 00 00 00  71
9e78 : 00 00 00 00 00 00 00 00  79
9e80 : 00 00 00 00 00 00 00 00  81
9e88 : 00 00 00 00 00 00 00 00  89
9e90 : 00 00 00 00 00 00 00 00  91
9e98 : 00 00 00 00 00 00 00 00  99
9ea0 : 00 00 00 00 00 00 00 00  a1
9ea8 : 00 00 00 00 00 00 00 00  a9
9eb0 : 00 00 00 00 00 00 00 00  b1
9eb8 : 00 00 00 00 00 00 00 00  b9
9ec0 : 00 00 00 00 00 00 00 00  c1
9ec8 : 00 00 00 00 00 00 00 00  c9
9ed0 : 00 00 00 00 00 00 00 00  d1
9ed8 : 00 00 00 00 00 00 00 00  d9
9ee0 : 00 00 00 00 00 00 00 00  e1
9ee8 : 00 00 00 00 00 00 00 00  e9
9ef0 : 00 00 00 00 00 00 47 4a  a2
9ef8 : 50 51 bf 2c 3b a2 59 dc  4e
9f00 : 00 80 84 84 84 83 80 00  58
9f08 : 00 80 82 84 83 80 80 00  b8
9f10 : 00 87 80 80 80 80 80 00  12
9f18 : 00 80 80 80 80 87 80 00  cf
9f20 : 00 87 80 80 80 80 80 00  22
9f28 : 00 80 82 81 80 80 80 00  47
9f30 : 00 80 82 81 80 80 80 00  4f
9f38 : 00 80 80 83 80 83 80 00  2f
9f40 : 00 80 84 84 83 80 80 00  70
9f48 : 00 82 84 84 84 83 80 00  a1
9f50 : a8 27 7e 00 00 00 00 00  2c
9f58 : 00 80 a0 80 a0 80 80 00  e1
9f60 : 00 80 80 c0 a0 80 80 00  e9
9f68 : 00 80 e0 80 e0 80 80 00  05
9f70 : 47 4a 50 51 bf 2c 3b a6  b2
9f78 : df bc 27 a2 7e 59 dc bb  71
9f80 : 00 b8 c4 c4 a8 fc 80 00  1a
9f88 : 00 80 80 80 fd 80 80 00  de
9f90 : 00 fc a8 c4 c4 b8 80 00  e5
9f98 : 00 b8 c4 c4 a8 fc 80 00  32
9fa0 : 00 fe c9 c9 b6 80 80 00  3c
9fa8 : 00 80 80 c0 80 80 80 00  2f
9fb0 : 00 80 80 d0 80 80 80 00  39
9fb8 : 00 a0 d5 d4 bd c0 80 00  fc
9fc0 : 00 b8 c5 c4 c5 b8 80 00  4a
9fc8 : 00 bd c0 c0 bd c0 80 00  d3
9fd0 : 00 80 81 80 80 80 80 00  8f
9fd8 : 00 80 80 c0 80 c0 80 00  61
9fe0 : 00 80 81 80 81 80 80 00  af
9fe8 : 00 bc c0 a0 fc 80 80 00  60
9ff0 : 00 b6 c9 c9 c9 b2 80 00  2b
9ff8 : 00 be dd e3 e3 be 80 00  81
```

Listing 1. »UMLAUT1«, eine verbesserte Druckroutine für Master-Text mit MPS 801/803. Bitte mit dem MSE (Seite 159) eingeben.

Werkzeugkasten – drei neue Erweiterungen zu Vizawrite 64

Die im 64'er-Magazin begonnene Serie der Zusatzprogramme für Vizawrite 64 wird hier fortgesetzt. Drei neue Erweiterungen präsentieren sich als Komplett-Paket für noch mehr Komfort bei der Textverarbeitung.

In Ausgabe 9/86 des 64'er-Magazins stellten wir Ihnen auf S. 156 ff. erstmals eine als »Vizaspell-Trick« bezeichnete Möglichkeit zur Erweiterung des bekannten Textprogramms Vizawrite 64 vor. Dabei geht es darum, daß sich im Texteditor von Vizawrite über die Tastenkombination ⟨CBM⟩ ⟨SHIFT RUN/STOP⟩, die üblicherweise dafür vorgesehen ist, die englische Rechtschreibhilfe Vizaspell 64 von Diskette nachzuladen, verschiedene Zusatzprogramme aktivieren lassen. Diese dürfen bis zu 6000 Byte ($5DC0 bis $752F) oder 23 Blöcke einer Diskette lang sein; der Programmname muß mit den Buchstaben »VIZA« beginnen. Die dadurch gebotenen Möglichkeiten sind enorm und im Grunde nur durch Fantasie und Geschick der Programmierer begrenzt. Dieser Trick funktioniert – wie sich leider herausgestellt hat – jedoch nicht bei der Modul-Version von Vizawrite.

Wir stellten Ihnen in diesem Zusammenhang bislang eine leistungsfähige Taschenrechner- und Kalender-Routine (Vizacalc in Ausgabe 4/87) sowie ein System zur Benutzung von Grafik-Schriften und zur Einbindung von Grafiken in Texte (Viza-Print-System in Ausgabe 6/87) vor. In diesem Heft finden Sie gleich fünf neue Erweiterungen, die übrigens alle von dem Autor des Viza-Print-Systems, Dieter Bayer, stammen.

Weil Vizawrite 64 bei der Tastenkombination ⟨CBM⟩ ⟨SHIFT RUN/STOP⟩ stets nur das erste auf der Diskette befindliche Programm mit dem Namen »VIZA*« sucht, müßte jedes dieser Teilprogramme auf einer eigenen Diskette gespeichert werden, was natürlich zu umständlich ist.



Bild 1. Zeichensätze gestalten mit VIZA-EDIT. Durch gute Menüführung wird die Bedienung erheblich erleichtert.

Wir bedienen uns daher eines einfachen Tricks, der bereits bei dem Viza-Print-System zur Anwendung kam und der ein mühsames Diskettenwechseln erübrigt: Wir rufen über ⟨CBM⟩ ⟨SHIFT RUN/STOP⟩ zunächst das Ladeprogramm »VIZA BOOTER« (Listing 1) auf, dessen Name im Directory selbstverständlich vor allen anderen Zusatzprogrammen stehen muß. »VIZA BOOTER« erlaubt uns dann das Nachladen der hier vorgestellten, aber auch anderer Vizawrite-Erweiterungen, die den eingangs erwähnten Regeln folgen.

»VIZA BOOTER« bietet uns zunächst ein Menü mit drei Punkten:

⟨F1⟩ Directory

Über diesen Menüpunkt wird das Inhaltsverzeichnis der Diskette auf dem Bildschirm ausgegeben. Die Anzeige läßt sich wie gewohnt mit ⟨CTRL⟩ verlangsamen; bei Druck auf

⟨CBM⟩ wird die Ausgabe bis zu einer erneuten Betätigung von ⟨CBM⟩ unterbrochen.

⟨F3⟩ Boot-Programm

Mit dieser Funktion wird das Aktivieren der einzelnen Vizawrite-Erweiterungen ermöglicht. Dabei erscheint zunächst im unteren Bildschirmteil die Liste der gültigen Programme (»VIZA*«) der eingelegten Diskette. Mit den Tasten ⟨CURSOR-aufwärts⟩ und ⟨CURSOR-abwärts⟩ läßt sich das gewünschte Programm heraussuchen; ⟨RETURN⟩ lädt und startet die gewählte Erweiterung. ⟨RUN/STOP⟩ bricht diesen Programmteil ab und führt zurück ins Hauptmenü.

⟨F8⟩ Return to Vizawrite

Wenn Sie den »VIZA BOOTER« versehentlich aufgerufen oder es sich anders überlegt haben, gelangen Sie von hier aus wieder zurück in den normalen Texteditor von Vizawrite.

Der »VIZA BOOTER« ist also lediglich ein Hilfsmittel und wird erst dann sinnvoll, wenn wir einige oder alle der nachfolgend beschriebenen Programme von Vizawrite 64 aus benutzen wollen. Selbstverständlich könnten diese Routinen aber auch direkt über ⟨CBM⟩ ⟨SHIFT RUN/STOP⟩ gestartet werden, allerdings - wie beschrieben - immer nur eine pro Diskette. Im einzelnen handelt es sich um folgende Programm-Module:

2) »VIZA-SQUEEZER« (Listing 2): Damit lassen sich Viza-Textfiles in eine platzsparende, handliche Form komprimieren.

4) »VIZA-EDIT« (Listing 3): Wenn Sie einmal den Bildschirmzeichensatz bei Vizawrite verändern möchten, bitte sehr, mit diesem Programm wird dies ein Kinderspiel.

5) »VIZA-SPELLER« (Listing 4): Durch diesen sehr einfach zu bedienenden Spell-Checker lehren Sie Ihrem Tippfehler-Teufel das Fürchten.

Viza-Squeezer — Datenfiles eng gepackt

Wer seine Viza-Textfiles möglichst platzsparend speichern möchte, erhält mit dem »VIZA-SQUEEZER« (Listing 2) einen praktischen und einfach zu bedienenden »Daten-Packer«, der die Blocklänge gegenüber dem Standardformat um etwa ein Drittel reduziert. Die auf diese Weise komprimierten Files sind allerdings über die normale Laderoutine von Vizawrite nicht mehr aufrufbar, sondern müssen stets über den kleinen Umweg dieses Hilfsprogramms geladen und wieder »entpackt« werden. Nach dem Starten des Programms erscheint ein Menü, das sich im wesentlichen von selbst erklärt. Trotzdem soll hier eine kurze Beschreibung der einzelnen Menüpunkte erfolgen:

⟨F1⟩ Directory

Anzeige des Directory der eingelegten Diskette. Mit der ⟨CTRL⟩-Taste kann das Scrollen verlangsamt und mit der ⟨CBM⟩-Taste angehalten werden.

⟨F3⟩ Load Text

Laden eines komprimierten Textes. Dabei wird zunächst das Directory der aktuellen Diskette eingelesen, und der Name des gewünschten Textes kann mit den Cursor-Tasten und ⟨RETURN⟩ ausgewählt werden. Die ⟨STOP⟩-Taste beendet diesen Menüpunkt vorzeitig.

⟨F5⟩ Save Text

Speichern eines Textes. Der Text wird beim Speichern automatisch komprimiert, wobei der Computer in der untersten Bildschirmzeile anzeigt, womit er gerade beschäftigt ist. Soll ein alter Text überschrieben werden, so ist dem Dateinamen ein »s:« voranzustellen. Um komprimierte Texte von anderen leichter unterscheiden zu können, empfiehlt sich die Verwendung eines Kürzels, wie etwa ».SQ«, am Namensende.

⟨F8⟩ Return to Vizawrite

Damit kehren Sie zurück ins Vizawrite-Hauptprogramm; einen mit »VIZA-SQUEEZER« geladenen Text können Sie nun wie gewohnt bearbeiten.

Viza-Edit — der Zeicheneditor

Wenn Ihnen der Bildschirmzeichensatz von Vizawrite 64 nicht gefällt, oder wenn Sie bestimmte Sonderzeichen (beispielsweise mathematische Symbole) vereinbaren wollen, so steht Ihnen mit »VIZA-EDIT« (Listing 3) ein sehr einfach zu bedienendes Hilfsmittel zur Verfügung. Sie können damit jedes der 256 Zeichen des CBM-Zeichensatzes neu definieren und diese geänderten Zeichensätze auf Diskette speichern. Anfänger sollten zunächst vorsichtig mit diesem Programm arbeiten, oder sich erst einmal den Muster-Zeichensatz (auf der Programmservice-Diskette) ansehen, weil man bei unsachgemäßer Behandlung auch ein heilloses Durcheinander der Zeichen schaffen kann. Wichtig ist, daß mit »VIZA-EDIT« lediglich die Bildschirm-, nicht aber die Druckausgabe der Zeichen beeinflußt wird.

Für das Laden und Starten gilt dasselbe wie bei den bereits beschriebenen Programmen. Nach dem Start erscheint ein Menü mit fünf Unterpunkten:

⟨F1⟩ - Display Directory
⟨F3⟩ - Load Zeichensatz
⟨F5⟩ - Save Zeichensatz
⟨F7⟩ - Edit Zeichensatz
⟨F8⟩ - Return to Vizawrite

Die Punkte ⟨F1⟩ und ⟨F8⟩ erklären sich von selbst (siehe auch bei »VIZA BOOTER«), so daß hier nur die restlichen Funktionen zu erläutern sind.

⟨F3⟩ - Load Zeichensatz

Damit wird ein auf Diskette gespeicherter Zeichensatz für Vizawrite 64 in den Arbeitsspeicher ab $3800 kopiert. Die Auswahl aus dem Inhaltsverzeichnis erfolgt wie bei »VIZA BOOTER«. Zur Demonstration dieser Funktion befindet sich auf der Programmservice-Diskette der Muster-Zeichensatz »ZEICHEN-DEMO«.

⟨F5⟩ - Save Zeichensatz

Der aktuelle Zeichensatz wird auf Diskette gespeichert. Dabei wird ein Zeichensatz gleichen Namens vor dem Speichern automatisch auf der Diskette gelöscht.

⟨F7⟩ - Edit Zeichensatz

Dieser Teil dient zum Editieren des Zeichensatzes. Beim Aufruf erscheint im unteren Bildschirmbereich ein weiteres Menü, außerdem eine 8 x 8 Punkte große Zeichenmatrix, wobei das jeweils aktuelle Zeichen zusätzlich in Normalgröße und mit der dazugehörigen Nummer links daneben steht (Bild 1). Die Unterpunkte des Menüs sollen kurz beschrieben werden:

⟨F1⟩ - Invertieren

Das aktuelle Zeichen (siehe Matrix) wird invertiert.

⟨F2⟩ - Automove an/aus

Wenn der Automove-Modus angeschaltet ist, wird der Cursor beim Setzen und Löschen von Punkten in der Zeichenmatrix immer in die Richtung weitergesetzt, in die man vorher den Cursor bewegt hat.

⟨F3⟩ - Zeichencode +
⟨F5⟩ - Zeichencode —

Hiermit kann im Zeichensatz stufenweise nach oben beziehungsweise nach unten »geblättert« werden. Man gelangt so rasch zu denjenigen Zeichen, die man verändern möchte.

⟨F7⟩ - Copy Character

Diese Funktion dient zum Kopieren eines Zeichens an eine andere Stelle im Zeichensatz. Es erscheint zuerst die Frage »Copy what ?«. Daraufhin wählt man das zu kopie-



rende Zeichen mit ⟨F3⟩ oder ⟨F5⟩ aus und übernimmt es mit ⟨RETURN⟩. Durch Drücken der ⟨RUN STOP⟩-Taste verläßt man den Copymodus vorzeitig. Nach Drücken der ⟨RETURN⟩-Taste fragt der Computer »Copy to where ?«. Jetzt sucht man das Zeichen aus, in dessen Matrix man das vorher bestimmte Zeichen kopieren will. Durch ⟨RETURN⟩ wird dies ausgeführt und der Kopiermodus wieder verlassen. Zur besseren Erkennung des Copy-Modus wird übrigens die Rahmenfarbe auf Hellgrau umgeschaltet. Weiterhin kann man im Copy-Modus durch Drücken der ⟨F1⟩-Taste das jeweils reverse Gegenstück des aktuellen Zeichens anwählen. Hierbei wird einfach zum Zeichencode der Wert 128 dazuaddiert. Auf diese Weise ist es sehr einfach möglich, reverse Zeichen zu erstellen.

⟨F8⟩ – Return to Menu
führt ins »VIZA-EDIT«-Hauptmenü zurück.

Viza-Speller – Tippfehler ade

Den Knüller dieses Viza-Werkzeugkastens haben wir uns für den Schluß aufgespart; dazu aber erst ein paar Vorbemerkungen. Bekanntlich gibt es zu Vizawrite 64 das Rechtschreib-Korrekturprogramm »Vizaspell«, auf dem ja der ganze Trick mit den vielen weiteren von uns vorgestellten Zusatzprogrammen beruht.

Leider eignet sich dieses kommerzielle Programm aber nur für Texte ohne Umlaute, da es für die englische Version von Vizawrite 64 geschrieben wurde. Die Anwendung von Vizaspell ist darum für deutschsprachige Texte mit Schwierigkeiten verbunden (Näheres dazu finden Sie in Ausgabe 6/86 des 64'er-Magazins). Schon einmal stellten wir Ihnen (in Ausgabe 11/1986) darum eine auch für die deutsche Version von Vizawrite 64 geeignete Rechtschreibhilfe vor. Bei »SPELL CHECK« handelte es sich um ein Programm, das unabhängig von Vizawrite gestartet wird. Das hier vorgestellte Programm »VIZA-SPELLER« (Listing 4) geht dagegen denselben Weg wie das englische Vizaspell, es wird über ⟨CBM⟩ ⟨SHIFT RUN/STOP⟩ oder über den »VIZA BOOTER« aufgerufen. Wenngleich dieser »Spell-Checker« nicht ganz so viele Funktionen hat wie das Vizaspell-Original, so gefiel uns diese Lösung doch so gut, daß wir sie Ihnen nicht vorenthalten wollten. »VIZA-SPELLER« überprüft einen im Arbeitsspeicher befindlichen Text anhand einer auf Diskette befindlichen Wortliste. Diese wird mit dem Hilfsprogramm »DICTIONARY.GEN« (Listing 5) erzeugt und ist zunächst leer, läßt sich aber mit Hilfe vorhandener Texte leicht auffüllen und an individuelle Bedürfnisse anpassen. Auf der Programmservice-Diskette befindet sich bereits eine solche Sammlung von deutschen Wörtern, die als Ausgangsbasis für ein derartiges elektronisches Wörterbuch dienen kann.

Die Arbeitsweise von »VIZA-SPELLER« ist im Prinzip recht einfach: Vor dem eigentlichen »Spell-Checking« wird von dem zu prüfenden Text erst einmal eine alphabetische Wortliste erzeugt. Diese Wortliste vergleicht das Programm anschließend mit seinem Wörterbuch. Unbekannte Wörter können in das Wörterbuch aufgenommen, ignoriert oder – falls es sich um falsch geschriebene Wörter handelt – für eine spätere Korrektur vorgemerkt werden. Dieser letzte Schritt muß von Hand vorgenommen werden. Dabei handelt es sich aber nur um eine kleine Mühe, die durch die volle Berücksichtigung des deutschen Zeichensatzes mehr als ausgeglichen wird.

Nach dem Laden von »VIZA-SPELLER« erscheint in der zweiten Bildschirmzeile ein kleines Menü, bestehend aus den Buchstaben »c, d, e, f, i, t, u, w, C, M, P« und »V«. Diese einzelnen Menüpunkte werden durch Druck auf die entsprechende Taste aufgerufen. Hinter den Buchstaben verbergen sich dabei folgende Funktionen:

c – Check

Dies ist die eigentliche Check-Funktion. Damit wird die Überprüfung der im Speicher befindlichen Wortliste (siehe unter Funktion »t«) des aktuellen Textes veranlaßt. Voraussetzung hierfür ist, daß sich im Diskettenlaufwerk eine Diskette mit einer Wortschatz-Bibliothek befindet. Sollte während des Korrekturvorganges ein unbekanntes, also (noch) nicht in der Wortschatzliste vorhandenes Wort gefunden werden, so unterbricht das Programm die Suche, und es erscheint eine entsprechende Mitteilung auf dem Bildschirm. Sie können dann entscheiden, ob das unbekannte Wort »gelernt«, also in die Liste aufgenommen, oder übergangen werden soll (Bild 2). Drücken Sie dazu ⟨J⟩ oder ⟨N⟩.



Bild 2. Der Viza-Speller in Aktion

Wollen Sie den Korrekturvorgang jedoch vorzeitig abbrechen, so ist dies mit ⟨RUN/STOP⟩ an diesen Haltepunkten möglich.

d – Directory

Es wird das Inhaltsverzeichnis der eingelegten Diskette angezeigt.

e – Wörter löschen

Wird einmal ein Wort fälschlicherweise übernommen, oder wird es aus anderen Gründen nicht mehr benötigt (beispielsweise ein nicht mehr benutzter Spezialausdruck oder Name), so kann es mit dieser Funktion wieder aus der Wortschatzliste gestrichen werden. Dazu geben Sie das jeweilige Wort oder zumindest den Wortanfang ein. Der Computer durchsucht nun den Wortschatz und gibt jedes Wort, das mit der gesuchten Zeichenfolge beginnt, auf dem Bildschirm aus. Dabei können Sie sich dann jeweils entscheiden, ob Sie das Wort löschen wollen oder nicht.

f – Suchen eines Wortes

Wollen Sie wissen, welche Wörter in Ihrem Verzeichnis enthalten sind, oder wollen Sie die Schreibweise eines bestimmten Ausdrucks nachsehen, so geben Sie bei dieser Funktion wie bei »E« den Anfang des gesuchten Wortes ein. Das Programm durchsucht dann die Datei nach Wörtern, die mit der entsprechenden Zeichenfolge beginnen und listet diese auf dem Bildschirm auf. (Ein Tip: Durch Eingabe nur eines Anfangsbuchstabens erhalten Sie die komplette Liste der jeweiligen Sub-Datei Ihres Gesamtverzeichnisses.) Für die Suche nach deutschen Sonderzeichen gelten folgende Regeln:

Gesuchtes Zeichen	Eingabe auf der Tastatur
ä	[(linke eckige Klammer)
ü	] (rechte eckige Klammer)
ö	£ (Pfund-Zeichen)
ß	π (Pi-Zeichen)

Die Eingabe der Umlaute muß im übrigen auch beim Menüpunkt »e« auf die gleiche Art vorgenommen werden.

i – Wort-Statistik

Es wird die Gesamtzahl aller Wörter im Text sowie die Zahl der unterschiedlichen Wörter angegeben.

t – Wortliste des Textes erstellen

Aus dem aktuellen Text wird hiermit eine Wortliste mit allen vorkommenden Wörtern erstellt. Diese Liste ist alphabetisch sortiert. Sie müssen diesen Menüpunkt stets aufrufen, bevor Sie einen Text mit Hilfe der Funktion »C« überprüfen können.

Die übrigen Menüpunkte bedeuten:

u – Unbekannte Wörter ausgeben

Nachdem Sie Ihr Dokument mit »c« oder »C« überprüft haben, können Sie sich mit diesem Menüpunkt die unbekannten Wörter auf dem Bildschirm darstellen lassen.

w – Anzeigen der Wortliste des aktuellen Textes

C – Wortliste überprüfen

Dieser Menüpunkt (<SHIFT C>) überprüft die Wortliste, ohne daß Sie dabei am Computer verweilen müssen. Es wird nur der Inhalt der Wortliste überprüft und die unbekannten Wörter markiert. Es erfolgt keine Abfrage, ob diese Wörter gelernt werden sollen oder nicht.

M – Unbekannte Wörter zufügen

Nach Überprüfung der Wortliste können Sie diese mit »M« in das Wörterverzeichnis einfügen.

P – Ausdruck unbekannter Wörter

Unbekannte Wörter können auf dem Drucker ausgegeben werden, um sie später im Text zu verbessern. Sie müssen darauf achten, daß sich der Drucker im Kleinschrift-Modus befindet. Umlaute werden sonst nicht korrekt ausgedruckt werden.

V – Rückkehr zu Vizawrite 64

Da bei diesem Programm Wert auf eine möglichst große Wortspeicherkapazität gelegt wurde, können – wie schon erwähnt – nur die einzelnen Wörter automatisch auf korrekte Schreibung überprüft werden. Die im Text befindlichen falschen Wörter müssen Sie aber anschließend von Hand korrigieren. Dazu gehen Sie folgendermaßen vor: Sie rufen nach dem Überprüfen der Wörter die Wortliste mit »w« auf. Alle Wörter, die nicht im Wortschatz vorgekommen sind, werden jetzt in der Liste invers dargestellt (Bild 3). Kontrollieren Sie nun einfach, welche dieser Wörter falsch geschrieben sind und merken Sie sich diese (in der falschen Schreibweise). Nach der Rückkehr in den Text können Sie dann mit ⟨CBM⟩ ⟨SHIFT F⟩ nach den entsprechenden unkorrekten Wörtern suchen und die Korrektur durchführen.

```
viza-speller v3.0              by Dieter Bayer
 c d e f i t u w C M P V

2   eines              1   entgegengesetz
2   einfach            5   enthalten
1   einfacher          5   enthält
1   einfügen           1   entscheidet
1   eingeben           3   epson
1   eingeblendet       2   er
1   eingehalten        1   erfolgt
1   eingeschalten      1   erhalten
4   eingestellt        1   erhöhe
3   einmal             5   erhöht
8   eins               1   erkläre
1   einstellung        2   erklärt
1   einzeln            1   erläutert
1   einzelnen          2   erniedrigt
2   else               1   eröffnet
3   ende               1   erreichen
1   endif              1   erreicht
          Bitte eine Taste drücken
```

Bild 3. VIZA.SPELLER-Wortliste: Unbekanntes wird invertiert gezeigt

Ebenfalls in diesem Sonderheft befindet sich ein Spell-Checker für unser Textverarbeitungsprogramm Master-Text (Seite 56). Die beiden Spell-Checker sind in bezug auf die Wortschatzdateien zueinander kompatibel. Dies bedeutet, daß Sie, falls Sie mit beiden Programmen arbeiten, sich nur ein einziges »Lexikon« anzulegen brauchen.

(Dieter Bayer/Dr. Rudolf Egg)

```
Name : vizabooter              5dbc 62eb
-----------------------------------------
5dbc : 53 ff 00 00 a2 00 bd 00   30
5dc4 : 04 9d 00 70 bd 00 05 9d   d0
5dcc : 00 71 bd 00 06 9d 00 72   26
5dd4 : bd 00 07 9d 00 73 ca d0   6f
5ddc : e5 20 4c 08 20 52 23 20   47
5de4 : 6a 08 20 16 5e a9 00 85   5b
5dec : c6 20 e4 ff f0 fb c9 85   1c
5df4 : d0 03 4c 1d 5e c9 86 d0   ed
5dfc : 03 4c 9b 5e c9 8c d0 e9   f0
5e04 : a9 00 85 c6 20 52 23 20   49
5e0c : 52 23 a2 03 20 49 08 4c   fe
5e14 : 3a 08 a9 e7 a0 61 4c 1e   3c
5e1c : ab 18 a2 0c a0 07 20 f0   a2
5e24 : ff 8e 9a 5e a9 01 20 c9   93
5e2c : 61 20 cf ff a5 90 d0 4c   4c
5e34 : 20 cf ff a5 90 d0 45 20   d5
5e3c : cf ff a6 90 d0 3e aa 20   b0
5e44 : cf ff a4 90 d0 36 20 cd   29
5e4c : bd 20 cf ff a6 90 d0 2c   98
5e54 : c9 20 d0 02 a9 a0 20 d2   67
5e5c : ff d0 ee a9 0d 20 d2 ff   d1
5e64 : ee 9a 5e ad 9a 5e c9 17   df
5e6c : d0 06 ce 9a 5e 20 16 61   48
5e74 : 18 ae 9a 5e a0 07 20 f0   fa
5e7c : ff 4c 2d 5e 20 cc ff a9   74
5e84 : 7e 20 c3 ff a9 00 85 c6   41
5e8c : a5 c6 f0 fc 4c dd 5d 24   e1
5e94 : 3a 56 49 5a 41 2a 00 a9   50
5e9c : 07 20 c9 61 a5 50 48 a5   9b
5ea4 : 51 48 a9 00 8d d5 60 20   cd
5eac : af 60 20 d8 60 18 a2 0c   18
5eb4 : a0 08 20 f0 ff 8e 9a 5e   1a
5ebc : a2 04 20 cf ff a5 90 d0   73
5ec4 : 55 ca d0 f6 20 cf ff a6   5f
5ecc : 90 d0 4b 20 d2 ff d0 f4   f6
5ed4 : a9 0d 20 d2 ff a2 04 20   cc
5edc : cf ff a5 90 d0 38 ca d0   c2
5ee4 : f6 20 cf ff a6 90 d0 2e   6d
5eec : c9 22 f0 02 d0 f3 a0 00   72
5ef4 : 20 cf ff a6 90 d0 1f c9   70
5efc : 22 f0 07 91 50 c8 c0 10   f9
5f04 : d0 ee ee d5 60 20 af 60   48
5f0c : 20 cf ff a8 a6 90 d0 06   67
5f14 : 98 d0 f5 4c d9 5e 20 cc   c6
5f1c : ff a9 7e 20 c3 ff ce d5   b7
5f24 : 60 ad d5 60 8d d7 60 c9   89
5f2c : ff d0 03 4c 1a 60 a9 ff   29
5f34 : 8d d5 60 a2 0a ee d5 60   48
5f3c : ee 9a 5e 8a 48 18 ae 9a   95
5f44 : 5e a0 0c 20 f0 ff 68 aa   ff
5f4c : 20 af 60 a0 00 b1 50 20   7f
5f54 : d2 ff c8 c0 10 d0 f6 a9   27
5f5c : 0d 20 d2 ff ad d5 60 cd   d4
5f64 : d7 60 f0 03 ca d0 ce a9   ca
5f6c : 00 8d d5 60 a2 0d 8e 9a   b6
5f74 : 5e 20 fa 60 a9 00 85 c6   eb
5f7c : 20 e4 ff f0 fb c9 03 d0   e8
5f84 : 03 4c 1a 60 c9 91 d0 03   b2
5f8c : 4c 34 60 c9 11 d0 03 4c   80
5f94 : 5e 60 c9 0d d0 de ad d5   9d
5f9c : 60 20 af 60 a0 00 b1 50   76
5fa4 : 99 24 60 f0 05 c8 c0 10   3f
5fac : d0 f4 98 a2 24 a0 60 20   7a
5fb4 : bd ff a2 27 bd 00 70 9d   d7
5fbc : 00 04 ca 10 f7 a9 08 aa   b5
5fc4 : a8 20 ba ff 20 c0 ff a2   78
5fcc : 08 20 c6 ff 20 cf ff 20   57
5fd4 : cf ff 20 cf ff c9 53 d0   e2
5fdc : 35 8d bc 5d 20 cf ff c9   c7
5fe4 : ff d0 2b 8d bd 5d 20 cf   af
5fec : ff 8d be 5d 20 cf ff 8d   a9
5ff4 : bf 5d a9 5d 85 51 a9 c0   83
5ffc : 85 50 a2 31 bd b9 62 9d   e6
6004 : 98 07 ad 21 d0 9d 98 db   c3
600c : ca 10 f1 4c 98 07 20 cc   c0
6014 : ff a9 08 20 c3 ff 68 85   d7
601c : 51 68 85 50 4c dd 5d 00   36
6024 : 20 20 20 20 20 20 20 20   24
602c : 20 20 20 20 20 20 20 20   2c
6034 : ad 9a 5e c9 0d f0 0c 20   c8
603c : fa 60 ce d5 60 ce 9a 5e   78
6044 : 4c 75 5f ad d5 60 f0 0f   1b
604c : ce d5 60 20 fa 60 20 7f   53
6054 : 61 20 93 60 4c 75 5f 4c   3d
605c : 78 5f ad 9a 5e c9 16 f0   b1
6064 : 14 ad d5 60 cd d7 60 f0   cf
606c : 23 ee d5 60 20 fa 60 ee   c1
6074 : 9a 5e 4c 75 5f ad d5 60   7b
607c : cd d7 60 f0 0f ee d5 60   eb
6084 : 20 fa 60 20 16 61 20 93   52
608c : 60 4c 75 5f 4c 78 5f ad   bd
6094 : d5 60 20 af 60 18 ae 9a   4e
609c : 5e a0 0c 20 f0 ff a0 00   e3
60a4 : b1 50 20 d2 ff c8 c0 10   49
60ac : d0 f6 60 a9 00 8d d6 60   cd
60b4 : ad d5 60 18 0a 2e d6 60   95
60bc : 0a 2e d6 60 0a 2e d6 60   cd
60c4 : 0a 2e d6 60 18 69 00 85   7f
60cc : 50 a9 65 6d d6 60 85 51   21
60d4 : 60 00 00 00 a9 00 aa 9d   b5
60dc : 00 65 9d 00 66 9d 00 67   18
60e4 : 9d 00 68 9d 00 69 9d 00   11
60ec : 6a 9d 00 6b 9d 00 6c 9d   59
60f4 : 00 6d ca d0 e2 60 ae 9a   99
60fc : 5e 20 f0 e9 20 24 ea a0   f4
6104 : 0a b1 d1 49 80 91 d1 ad   bc
610c : 86 02 91 f3 c8 c0 1e d0   23
6114 : f0 60 a5 ac 48 a5 ad 48   2c
611c : a5 ae 48 a5 af 48 a2 0c   bf
6124 : ad 8d 02 c9 02 d0 0a ad   7c
612c : 8d 02 d0 fb ad 8d 02 f0   9f
6134 : fb e8 20 f0 e9 e0 16 b0   29
```

```
613c : 0c bd f1 ec 85 ac b5 da  8b
6144 : 20 c8 e9 30 ec 20 6a 61  85
614c : ad 8d 02 c9 04 d0 0a a2  ae
6154 : 00 a8 ea 88 d0 fc ca d0  36
615c : f9 68 85 af 68 85 ae 68  1f
6164 : 85 ad 68 85 ac 60 a0 27  29
616c : 20 f0 e9 20 24 ea a9 60  84
6174 : 91 d1 ad 86 02 91 f3 88  b8
617c : 10 f4 60 a5 ac 48 a5 ad  d2
6184 : 48 a5 ae 48 a5 af 48 a2  92
618c : 16 ca 20 f0 e9 e0 0c f0  e5
6194 : 0c bd f1 ec 85 ac b5 da  e3
619c : 20 b2 61 30 ec e8 20 6a  df
61a4 : 61 68 85 af 68 85 ae 68  cf
61ac : 85 ad 68 85 ac 60 29 03  4b
61b4 : 09 04 85 ad 20 e0 e9 a0  c8
61bc : 27 b1 d1 91 ac b1 f3 91  ae
61c4 : ae 88 10 f5 60 a2 93 a0  24
61cc : 5e 20 bd ff a9 7e a2 08  d3
61d4 : a0 00 20 ba ff 20 c0 ff  d8
61dc : a2 7e 20 c6 ff 20 cf ff  de
61e4 : 4c cf ff 13 12 d6 49 5a  2c
61ec : 41 42 4f 4f 54 45 52 20  05
61f4 : 20 20 20 20 20 20 57 52  36
61fc : 49 54 54 45 4e 20 42 59  cf
6204 : 20 c4 49 45 54 45 52 20  7a
620c : c2 41 59 45 52 92 11 1d  a6
6214 : 1d c0 c0 c0 c0 c0 c0 c0  70
621c : c0 c0 c0 c0 c0 c0 c0 c0  1b
6224 : c0 c0 c0 c0 c0 c0 c0 c0  23
622c : c0 c0 c0 c0 c0 c0 c0 c0  2b
6234 : c0 c0 c0 c0 c0 0d 1d 1d  c0
623c : 1d 1d 1d 1d 1d 46 31 a0  dd
6244 : 2d a0 c4 49 53 50 4c 41  87
624c : 59 20 c4 49 52 45 43 54  15
6254 : 4f 52 59 0d 1d 1d 1d 1d  2e
625c : 1d 1d 1d 46 33 a0 2d a0  46
6264 : c2 4f 4f 54 20 d0 52 4f  9d
626c : 47 52 41 4d 4d 0d 1d 1d  c2
6274 : 1d 1d 1d 1d 1d 46 38 a0  31
627c : 2d a0 d2 45 54 55 52 4e  2c
6284 : 20 54 4f 20 d6 49 5a 41  4a
628c : 57 52 49 54 45 0d 1d 1d  55
6294 : c0 c0 c0 c0 c0 c0 c0 c0  93
629c : c0 c0 c0 c0 c0 c0 c0 c0  9b
62a4 : c0 c0 c0 c0 c0 c0 c0 c0  a3
62ac : c0 c0 c0 c0 c0 c0 c0 c0  ab
62b4 : c0 c0 c0 c0 00 20 cf ff  5d
62bc : 4d be 5d 4d bf 5d a6 90  0c
62c4 : d0 0c a0 00 91 50 e6 50  9a
62cc : d0 02 e6 51 d0 e7 a0 00  50
62d4 : 91 50 20 cc ff a9 08 20  dd
62dc : c3 ff 68 85 51 68 85 50  79
62e4 : a9 30 85 01 4c c0 5d a0  a8
```

Listing 1. »VIZA BOOTER«. Das Programm dient zum menügesteuerten Aufruf der Vizawrite-Hilfsprogramme. Bitte mit dem MSE (Seite 159) eingeben.

```
Name : viza-squeezer        5dbc 6ac7
-----------------------------------
5dbc : 53 ff 00 00 a2 00 b5 00  10
5dc4 : 9d c6 6a ca d0 f8 20 4c  a6
5dcc : 08 20 3a 08 20 52 23 20  d5
5dd4 : e8 5e 20 3d 5e a9 12 20  57
5ddc : d2 ff 18 a2 00 a0 18 20  ae
5de4 : f0 ff a2 00 bd 8a 0e c9  78
5dec : 60 f0 08 20 d2 ff e8 e0  5d
5df4 : 10 d0 f1 a9 92 20 d2 ff  93
5dfc : a9 00 85 c6 20 e4 ff f0  ea
5e04 : fb c9 85 d0 03 4c 6d 5e  64
5e0c : c9 86 d0 03 4c 8a 64 c9  eb
5e14 : 87 d0 03 4c d1 62 c9 8c  be
5e1c : d0 e2 a2 00 bd c6 6a 95  ed
5e24 : 00 ca d0 f8 20 52 23 20  3e
5e2c : 52 23 20 4c 08 a9 00 85  7a
5e34 : c6 a2 03 20 49 08 4c 3a  8b
5e3c : 08 a9 ed a0 61 8d 62 5e  71
5e44 : 8c 63 5e 20 61 5e aa 30  32
5e4c : 1f 20 61 5e a8 18 20 f0  4d
5e54 : ff 20 61 5e aa f0 ec 20  ae
5e5c : d2 ff 4c 55 5e ad ff ff  3f
5e64 : ee 62 5e d0 03 ee 63 5e  27
5e6c : 60 18 a2 0e a0 07 20 f0  e7
5e74 : ff 8e e2 5e 20 bb 61 20  e4
5e7c : cf ff a5 90 d0 4c 20 cf  56
5e84 : ff a5 90 d0 45 20 cf ff  29
5e8c : a6 90 d0 3e aa 20 cf ff  61
5e94 : a4 90 d0 36 20 cd bd 20  23
5e9c : cf ff a6 90 d0 2c c9 20  fc
5ea4 : d0 02 a9 a0 20 d2 ff d0  2e
5eac : ee a9 0d 20 d2 ff ee e2  65
5eb4 : 5e ad e2 5e c9 17 d0 06  12
5ebc : ce e2 5e 20 0d 61 18 ae  31
5ec4 : e2 5e a0 07 20 f0 ff 4c  00
5ecc : 7b 5e 20 cc ff a9 7e 20  9f
5ed4 : c3 ff a9 00 85 c6 a5 c6  b4
5edc : f0 fc 4c ca 5d 24 00 a2  f3
5ee4 : 30 86 01 60 a2 37 86 01  a4
5eec : 60 20 bb 61 a9 00 8d cc  e2
5ef4 : 60 20 a6 60 20 cf 60 18  4c
5efc : a2 0e a0 08 20 f0 ff 8e  75
5f04 : e2 5e a2 04 20 cf ff a5  0a
5f0c : 90 d0 55 ca d0 f6 20 cf  98
5f14 : ff a6 90 d0 4b 20 d2 ff  a5
5f1c : d0 f4 a9 0d 20 d2 ff a2  50
5f24 : 04 20 cf ff a5 90 d0 38  bf
5f2c : ca d0 f6 20 cf ff a6 90  d9
5f34 : d0 2e c9 22 f0 02 d0 f3  1c
5f3c : a0 00 20 cf ff a6 90 d0  f7
5f44 : 1f c9 22 f0 07 91 50 c8  be
5f4c : c0 10 d0 ee ee cc 60 20  3d
5f54 : a6 60 20 cf ff a8 a6 90  2d
5f5c : d0 06 98 d0 f5 4c 23 5f  7c
5f64 : 20 cc ff a9 7e 20 c3 ff  17
5f6c : ce cc 60 ad cc 60 8d ce  12
5f74 : 60 c9 ff d0 03 4c 13 60  72
5f7c : a9 ff 8d cc 60 a2 08 ee  3b
5f84 : cc 60 ee e2 5e 8a 48 18  24
5f8c : ae e2 5e a0 0c 20 f0 ff  dc
5f94 : 68 aa 20 a6 60 a0 00 b1  9d
5f9c : 50 20 db 61 20 d2 ff c8  49
5fa4 : c0 10 d0 f3 a9 0d 20 d2  48
5fac : ff ad cc 60 cd ce 60 f0  78
5fb4 : 03 ca d0 cb a9 00 8d cc  34
5fbc : 60 a2 0f 8e e2 5e 20 f1  88
5fc4 : 60 a9 00 85 c6 20 e4 ff  ab
5fcc : f0 fb c9 03 d0 03 4c 13  09
5fd4 : 60 c9 91 d0 03 4c 28 60  8b
5fdc : c9 11 d0 03 4c 52 60 c9  2f
5fe4 : 0d d0 de ad cc 60 20 a6  64
5fec : 60 a0 00 b1 50 99 18 60  c6
5ff4 : f0 05 c8 c0 10 d0 f4 98  3e
5ffc : a2 18 a0 60 20 bd ff a9  22
6004 : 08 aa a0 62 20 ba ff 20  ee
600c : c0 ff a2 08 4c c6 ff 68  41
6014 : 68 4c ca 5d 31 32 33 34  da
601c : 35 36 37 38 39 30 31 32  7f
6024 : 33 34 35 36 ad e2 5e c9  84
602c : 0f f0 0c 20 f1 60 ce cc  b1
6034 : 60 ce e2 5e 4c c2 5f ad  33
603c : cc 60 f0 0f ce cc 60 20  6b
6044 : f1 60 20 71 61 20 87 60  92
604c : 4c c2 5f 4c c5 5f ad e2  2f
6054 : 5e c9 16 f0 14 ad cc 60  dd
605c : cd ce 60 f0 23 ee cc 60  64
6064 : 20 f1 60 ee e2 5e 4c c2  4b
606c : 5f ad cc 60 cd ce 60 f0  98
6074 : 0f ee cc 60 20 f1 60 20  8d
607c : 0d 61 20 87 60 4c c2 5f  65
6084 : 4c c5 5f ad cc 60 20 a6  de
608c : 60 18 ae e2 5e a0 0c 20  5c
6094 : f0 ff a0 00 b1 50 20 db  82
609c : 61 20 d2 ff c8 c0 10 d0  36
60a4 : f3 60 a9 00 8d cd 60 ad  56
60ac : cc 60 18 0a 2e cd 60 0a  d7
60b4 : 2e cd 60 0a 2e cd 60 0a  09
60bc : 2e cd 60 18 69 00 85 50  39
60c4 : a9 6c 6d cd 60 85 51 60  f1
60cc : 00 00 00 a9 00 aa 9d 00  cd
60d4 : 6c 9d 00 6d 9d 00 6e 9d  8b
60dc : 00 6f 9d 00 70 9d 00 71  d2
60e4 : 9d 00 72 9d 00 73 9d 00  e4
60ec : 74 ca d0 e2 60 ae e2 5e  19
60f4 : 20 f0 e9 20 24 ea a0 0a  3b
60fc : b1 d1 49 80 91 d1 ad 86  64
6104 : 02 91 f3 c8 c0 1e d0 f0  07
610c : 60 a5 ac 48 a5 ad 48 a5  a7
6114 : ae 48 a5 af 48 a2 0e ad  73
611c : 8d 02 c9 02 d0 05 ad 8d  64
6124 : 02 d0 fb e8 20 f0 e9 e0  9d
612c : 16 b0 0c bd f1 ec 85 ac  4b
6134 : b5 da 20 c8 e9 30 ec 20  8c
613c : 5c 61 ad 8d 02 c9 04 d0  86
6144 : 0a a2 00 a8 ea 88 d0 fc  e5
614c : ca d0 f9 68 85 af 68 85  8c
6154 : ae 68 85 ad 68 85 ac 60  73
615c : a0 27 20 f0 e9 20 24 ea  bc
6164 : a9 60 91 d1 ad 86 02 91  16
616c : f3 88 10 f4 60 a5 ac 48  bc
6174 : a5 ad 48 a5 ae 48 a5 af  da
617c : 48 a2 16 ca 20 f0 e9 e0  e7
6184 : 0e f0 0c bd f1 ec 85 ac  bb
618c : b5 da 20 a4 61 30 ec e8  68
6194 : 20 5c 61 68 85 af 68 85  ca
619c : ae 68 85 ad 68 85 ac 60  bb
61a4 : 29 03 09 04 85 ad 20 e0  1a
61ac : e9 a0 27 b1 d1 91 ac b1  a5
61b4 : f3 91 ae 88 10 f5 60 a9  b2
61bc : 01 a2 e1 a0 5e 20 bd ff  79
61c4 : a9 7e a2 08 a0 00 20 ba  56
61cc : ff 20 c0 ff a2 7e 20 c6  37
61d4 : ff 20 cf ff 4c cf ff f0  fc
61dc : 0c c9 20 90 09 c9 80 90  e9
61e4 : 04 c9 a0 90 01 60 a9 2e  1d
61ec : 60 00 00 12 d6 49 5a 41  32
61f4 : 2d d3 51 55 45 45 5a 45  7c
61fc : 52 20 31 2e 32 20 20 ce  b3
6204 : 41 4d 45 3a 20 20 20 20  48
620c : 20 20 20 20 20 20 20 20  0c
6214 : 20 20 20 20 00 02 02 92  8e
621c : c0 c0 c0 c0 c0 c0 c0 c0  1b
6224 : c0 c0 c0 c0 c0 c0 c0 c0  23
622c : c0 c0 c0 c0 c0 c0 c0 c0  2b
6234 : c0 c0 c0 c0 c0 c0 c0 c0  33
623c : c0 c0 c0 c0 00 04 06 46  69
6244 : 31 a0 2d a0 c4 49 52 45  8f
624c : 43 54 4f 52 59 00 06 06  91
6254 : 46 33 a0 2d a0 cc 4f 41  32
625c : 44 20 d4 45 58 54 00 08  c6
6264 : 06 46 35 a0 2d a0 d3 41  98
626c : 56 45 20 d4 45 58 54 00  70
6274 : 0a 06 46 38 a0 2d a0 d2  b6
627c : 45 54 55 52 4e 20 54 4f  61
6284 : 20 d6 49 5a 41 57 52 49  58
628c : 54 45 00 0c 02 c0 c0 c0  af
6294 : c0 c0 c0 c0 c0 c0 c0 c0  93
629c : c0 c0 c0 c0 c0 c0 c0 c0  9b
62a4 : c0 c0 c0 c0 c0 c0 c0 c0  a3
62ac : c0 c0 c0 c0 c0 c0 c0 c0  ab
62b4 : c0 00 ff 7c 7b 7a 79 78  66
62bc : 76 65 a2 37 86 01 20 cf  05
62c4 : ff a6 90 d0 04 18 4c e3  4e
62cc : 5e 38 4c e3 5e 18 a2 0e  23
62d4 : a0 07 20 f0 ff a9 eb a0  5c
62dc : 68 20 1e ab a9 12 a2 0e  23
62e4 : a0 0e 20 3c 6a 90 03 4c  eb
62ec : ca 5d a2 f4 a0 68 20 bd  f5
62f4 : ff ad f4 68 c9 53 d0 17  bd
62fc : ad f5 68 c9 3a d0 10 a9  b5
6304 : 08 aa a0 0f 20 ba ff 20  83
630c : c0 ff a9 08 20 c3 ff a9  ab
6314 : 08 aa a0 01 20 ba ff 20  d2
631c : c0 ff b0 1d a2 08 20 c9  2a
6324 : ff b0 16 a6 90 d0 12 a9  01
632c : 56 20 d2 ff a6 90 d0 09  8b
6334 : a9 53 20 d2 ff a6 90 f0  43
633c : 03 4c 7f 64 a5 2b 85 8b  b3
6344 : a5 2c 85 8c 18 a5 2b 6d  28
634c : 05 0b 8d 38 69 a5 2c 6d  91
6354 : 06 0b 8d 39 69 ad 44 0b  95
635c : 85 fd ad 45 0b 85 fe 20  0d
6364 : cf 67 78 a0 00 a9 ff 91  89
636c : fd 20 e3 5e 20 1d 65 20  ff
6374 : e8 5e 58 20 db 67 20 ae  7c
637c : 65 a5 2b 85 8b a5 2c 85  d1
6384 : 8c 20 e7 67 a2 00 bd 7b  1f
638c : 0e 20 d2 ff e8 e0 46 d0  af
6394 : f5 a0 00 a2 30 86 01 b1  cc
639c : 8b a2 37 86 01 c9 ff d0  17
63a4 : 03 4c 7c 64 c9 00 f0 16  05
63ac : c9 1b 90 14 c9 41 90 0e  ae
63b4 : c9 5b 90 0c a2 06 dd b7  12
```

Listing 2. »VIZA-SQUEEZER«. Damit lassen sich Viza-Textfiles in eine platzsparende, handliche Form komprimieren. Bitte mit dem MSE (Seite 159) eingeben.

```
63bc : 62 f0 05 ca 10 f8 38 24   23
63c4 : 18 90 4f 8d f3 68 c8 f0   31
63cc : 0f a2 30 86 01 b1 8b a2   1a
63d4 : 37 86 01 cd f3 68 f0 ee   6c
63dc : 88 c0 03 90 1d c8 18 98   41
63e4 : 65 8b 85 8b 90 02 e6 8c   af
63ec : a9 f0 20 d2 ff 98 20 d2   5b
63f4 : ff ad f3 68 20 d2 ff 4c   05
63fc : 95 63 a0 00 a2 30 86 01   33
6404 : b1 8b a2 37 86 01 20 d2   a1
640c : ff e6 8b d0 02 e6 8c 4c   9d
6414 : 97 63 a2 30 86 01 b1 8b   5a
641c : a2 37 86 01 c9 00 f0 16   a8
6424 : c9 1b 90 14 c9 41 90 0e   26
642c : c9 5b 90 0c a2 06 dd b7   8a
6434 : 62 f0 05 ca 10 f8 38 24   9b
643c : 18 b0 08 99 06 69 c8 c0   32
6444 : 28 d0 cf c9 20 d0 04 99   cd
644c : 06 69 c8 84 02 c0 03 b0   5d
6454 : 21 a0 00 a2 30 86 01 b1   b8
645c : 8b a2 37 86 01 20 d2 ff   33
6464 : c8 c4 02 d0 ee 18 98 65   06
646c : 8b 85 8b 90 02 e6 8c 4c   d1
6474 : 95 63 20 c5 66 4c 95 63   61
647c : 20 d2 ff 20 cc ff a9 08   8d
6484 : 20 c3 ff 4c ca 5d 20 ed   03
648c : 5e 20 be 62 b0 0d c9 56   3d
6494 : d0 09 20 be 62 b0 04 c9   18
649c : 53 f0 0b 20 cc ff a9 08   b2
64a4 : 20 c3 ff 4c ca 5d 20 be   c5
64ac : 62 8d 44 0b 85 fb 20 be   7e
64b4 : 62 8d 45 0b 85 fc a5 2b   bd
64bc : 85 8b a5 2c 85 8c 20 7a   28
64c4 : 67 20 b3 67 20 b2 68 ee   2c
64cc : c0 07 20 be 62 c9 ff f0   46
64d4 : 25 c9 f0 d0 06 20 96 68   c1
64dc : 4c cb 64 c9 60 90 0a c9   a6
64e4 : 80 b0 06 20 36 68 4c cb   b1
64ec : 64 a0 00 91 8b e6 8b d0   92
64f4 : 02 e6 8c 4c cb 64 a0 00   78
64fc : 91 8b 20 cc ff a9 08 20   a2
6504 : c3 ff 38 a5 2d ed 7f 0e   e6
650c : 8d 42 0b a5 2e ed 80 0e   a2
6514 : 8d 43 0b 20 e6 2e 4c ca   b0
651c : 5d a9 c6 85 57 a9 6b 85   2c
6524 : 58 ee c0 07 20 39 65 b0   c7
652c : 0b 20 30 66 b0 f3 20 78   3c
6534 : 66 4c 25 65 60 a0 00 b1   25
653c : 8b c9 00 f0 16 c9 1b 90   07
6544 : 14 c9 41 90 0e c9 5b 90   5d
654c : 0c a2 06 dd b7 62 f0 05   43
6554 : ca 10 f8 38 24 18 90 0d   cb
655c : c9 ff f0 4c e6 8b d0 02   fc
6564 : e6 8c 4c 3b 65 b1 8b c9   b0
656c : 00 f0 16 c9 1b 90 14 c9   bd
6574 : 41 90 0e c9 5b 90 0c a2   6a
657c : 06 dd b7 62 f0 05 ca 10   2d
6584 : f8 38 24 18 b0 08 99 06   62
658c : 69 c8 c0 28 d0 d7 c9 20   c2
6594 : d0 04 99 06 69 c8 84 02   80
659c : 18 98 65 8b 85 8b 90 02   c6
65a4 : e6 8c c0 03 90 8f 18 60   07
65ac : 38 60 ad 44 0b 85 8b 20   54
65b4 : d2 ff ad 45 0b 85 8c 20   e9
65bc : d2 ff a0 00 a2 30 86 01   7e
65c4 : b1 8b a2 37 86 01 c9 ff   62
65cc : f0 12 20 e3 65 90 ed 20   1d
65d4 : d2 ff e6 8b d0 02 c6 8c   a3
65dc : 4c c0 65 60 4c d2 ff a2   8e
65e4 : 00 dd 3c 69 f0 08 e8 e8   d4
65ec : e0 fe d0 f5 38 60 e8 c8   fa
65f4 : 86 03 a2 30 86 01 b1 8b   f9
65fc : a2 37 86 01 a6 03 dd 3c   6e
6604 : 69 d0 17 8a 4a 09 80 20   1c
660c : d2 ff a0 02 18 98 65 8b   39
6614 : 85 8b 90 02 e6 8c 18 a0   38
661c : 00 60 88 86 03 a2 30 86   52
```

```
6624 : 01 b1 8b a2 37 86 01 a6   2e
662c : 03 4c eb 65 ad 44 0b 85   31
6634 : fb ad 45 0b 85 fc a9 c6   2d
663c : 85 59 a9 6b 85 5a a0 00   f3
6644 : b1 fb 30 2e d9 06 69 f0   1a
664c : 15 a0 00 b1 59 18 65 fb   cb
6654 : 85 fb 90 02 e6 fc e6 59   e0
665c : d0 e4 e6 5a d0 e0 c8 c4   64
6664 : 02 f0 09 b1 fb d9 06 69   d0
666c : f0 f4 d0 dd b1 fb d0 d9   b8
6674 : 38 24 18 60 38 ad 38 69   75
667c : e5 fd 8d 3a 69 ad 39 69   c6
6684 : e5 fe 8d 3b 69 d0 08 ad   4c
668c : 3a 69 c9 50 b0 01 60 a4   d5
6694 : 02 a9 00 91 fd c8 84 03   db
669c : a9 ff 91 fd 88 88 b9 06   29
66a4 : 69 91 fd 88 10 f8 a4 03   c8
66ac : 18 98 65 fd 85 fd 90 02   b8
66b4 : e6 fe a0 00 a6 02 e8 8a   75
66bc : 91 57 e6 57 d0 02 e6 58   07
66c4 : 60 20 19 67 90 3c a0 00   d5
66cc : a2 30 86 01 b1 8b a2 37   b8
66d4 : 86 01 c9 00 f0 16 c9 1b   6a
66dc : 90 14 c9 41 90 0e c9 5b   68
66e4 : 90 0c a2 06 dd b7 62 f0   eb
66ec : 05 ca 10 f8 38 24 18 b0   e0
66f4 : 06 20 d2 ff c8 d0 d1 18   49
66fc : 98 65 8b 85 8b 90 02 e6   ed
6704 : 8c 60 18 a9 60 6d e9 68   e5
670c : 20 d2 ff ad e8 68 20 d2   43
6714 : ff a4 02 d0 e2 a9 00 a2   c1
671c : 30 86 01 8d e8 68 8d e9   5d
6724 : 68 ad 44 0b 85 50 ad 45   f1
672c : 0b 85 51 a9 c6 85 59 a9   d5
6734 : 6b 85 5a a0 00 b1 50 30   3c
673c : 37 d9 06 69 f0 1d a0 00   89
6744 : b1 59 18 65 50 85 50 90   e8
674c : 02 e6 51 e6 59 d0 02 e6   e4
6754 : 5a ee e8 68 d0 dd ee e9   f8
675c : 68 d0 d8 c8 b1 50 f0 07   eb
6764 : d9 06 69 f0 f6 d0 d7 c4   98
676c : 02 f0 03 88 d0 18 24 18   f4
6774 : 38 a2 37 86 01 60 a0 00   32
677c : 20 be 62 c9 80 90 22 c9   76
6784 : ff f0 29 29 7f 0a aa bd   d9
678c : 3c 69 91 fb c8 e8 bd 3c   a4
6794 : 69 91 fb c8 18 98 65 fb   b2
679c : 85 fb 90 02 e6 fc 4c 7a   00
67a4 : 67 91 fb e6 fb d0 02 e6   cc
67ac : fc 4c 7c 67 91 fb 60 a0   96
67b4 : 00 20 be 62 99 7b 0e c8   ff
67bc : c0 46 d0 f5 60 a2 00 86   ba
67c4 : c6 20 e8 5e a6 c6 f0 fc   ff
67cc : 4c e3 5e a2 16 bd f3 67   e4
67d4 : 9d c8 07 ca 10 f7 60 a2   78
67dc : 16 bd 0a 68 9d c8 07 ca   32
67e4 : 10 f7 60 a2 16 bd 1f 68   f9
67ec : 9d c8 07 ca 10 f7 60 2d   a5
67f4 : 20 47 45 4e 45 52 41 54   68
67fc : 49 4e 47 20 57 4f 52 44   04
6804 : 4c 49 53 54 20 2d 60 60   02
680c : 2d 20 53 41 56 49 4e 47   be
6814 : 20 57 4f 52 44 4c 49 53   70
681c : 54 20 2d 60 60 60 60 2d   bd
6824 : 20 53 41 56 49 4e 47 20   6d
682c : 54 45 58 54 20 2d 60 60   71
6834 : 60 60 38 e9 60 8d e9 68   fa
683c : ad 44 0b 85 fb ad 45 0b   d7
6844 : 85 fc a9 c6 85 59 a9 6b   2b
684c : 85 5a 20 be 62 8d e8 68   e5
6854 : ad e9 68 0d e8 68 f0 24   8f
685c : a0 00 b1 59 18 65 fb 85   3b
6864 : fb 90 02 e6 fc e6 59 d0   13
686c : 02 e6 5a 38 ad e8 68 e9   17
6874 : 01 8d e8 68 b0 da ce e9   74
687c : 68 4c 54 68 a0 00 b1 fb   f5
6884 : f0 05 91 8b c8 d0 f7 18   f0
```

```
688c : 98 65 8b 85 8b 90 02 e6   7d
6894 : 8c 60 20 be 62 85 03 20   cf
689c : be 62 a0 00 91 8b c8 c4   d6
68a4 : 03 d0 f9 18 98 65 8b 85   7f
68ac : 8b 90 02 e6 8c 60 ad 44   e8
68b4 : 0b 85 fb ad 45 0b 85 fc   f3
68bc : a9 c6 85 57 a9 6b 85 58   d1
68c4 : a0 00 b1 fb 30 1d c8 b1   c3
68cc : fb d0 fb c8 98 a0 00 91   f9
68d4 : 57 18 65 fb 85 fb 90 02   8f
68dc : e6 fc e6 57 d0 02 e6 58   4e
68e4 : 4c c6 68 60 00 00 00 ce   57
68ec : 41 4d 45 a0 3a a0 00 00   e2
68f4 : 31 32 33 34 35 36 37 38   e4
68fc : 39 30 31 32 33 34 35 36   f6
6904 : 37 38 31 32 33 34 35 36   00
690c : 37 38 39 30 31 32 33 34   8e
6914 : 35 36 37 38 39 30 31 32   77
691c : 33 34 35 36 37 38 39 30   f8
6924 : 31 32 33 34 35 36 37 38   14
692c : 39 30 31 32 33 34 35 36   26
6934 : 37 38 39 30 00 00 00 00   dc
693c : 20 00 02 05 03 08 04 05   08
6944 : 05 08 05 09 05 0c 05 0e   91
694c : 05 12 05 13 05 14 07 05   15
6954 : 08 05 09 03 09 05 09 0e   7b
695c : 09 14 04 09 04 12 04 15   9d
6964 : 05 02 05 03 05 04 05 06   9d
696c : 12 15 13 03 13 05 13 14   fc
6974 : 14 05 15 0e 01 0c 01 0d   a1
697c : 01 12 04 01 07 12 09 12   f1
6984 : 0d 0d 0f 12 12 01 12 09   a2
698c : 12 0f 15 0e 01 14 01 15   0c
6994 : 03 0b 05 07 09 0c 0c 01   62
699c : 0f 0c 0f 0e 0f 10 15 13   23
69a4 : 15 14 1a 15 0c 0c 0e 0e   62
69ac : 13 13 09 1a 0d 0f 0e 05   5a
69b4 : 0f 0d 12 14 13 10 1a 01   6d
69bc : 01 03 01 04 03 05 17 05   be
69c4 : 03 15 05 01 05 18 07 08   f1
69cc : 09 07 09 13 0c 05 0c 09   29
69d4 : 0c 0f 0c 19 0e 07 0f 03   e9
69dc : 0f 15 0f 11 10 0c 11 15   2c
69e4 : 12 05 13 15 14 01 14 08   8a
69ec : 14 09 15 0d 15 12 01 02   56
69f4 : 05 0d 06 06 06 14 09 01   e9
69fc : 09 0d 0c 14 0d 15 10 10   eb
6a04 : 12 0e 1a 17 0e 14 0f 17   73
6a0c : 01 0e 06 78 12 78 0d 76   ab
6a14 : 17 65 01 08 01 0b 03 0f   b2
6a1c : 07 14 08 14 0b 14 13 0f   6e
6a24 : 14 1a 16 05 01 07 01 13   de
6a2c : 01 19 17 01 06 15 07 01   c7
6a34 : 08 01 09 06 13 09 ff ff   39
6a3c : 8d b6 6a 18 20 f0 ff a9   9f
6a44 : 00 85 d4 8d b5 6a 85 cc   4c
6a4c : 20 e4 ff f0 fb c9 03 f0   f8
6a54 : 5b c9 0d f0 3d a2 0e dd   d2
6a5c : b7 6a f0 ec ca 10 f8 c9   c7
6a64 : 14 d0 1b ae b5 6a f0 e0   b1
6a6c : a4 d3 b1 d1 29 7f 20 a6   fd
6a74 : 6a c6 d3 a9 60 20 a6 6a   e2
6a7c : ce b5 6a 4c 4c 6a ae b5   87
6a84 : 6a ec b6 6a f0 c2 ee b5   ab
6a8c : 6a 9d f4 68 20 d2 ff 4c   40
6a94 : 4c 6a a0 01 84 cc a5 ce   40
6a9c : 29 7f 20 a6 6a 18 ad b5   eb
6aa4 : 6a 60 a0 00 84 cf ae 87   f7
6aac : 02 4c 13 ea 20 96 6a 38   c7
6ab4 : 60 00 00 93 13 11 91 9d   c2
6abc : 1d 94 85 89 86 8a 87 8b   a8
6ac4 : 88 8c 00 b1 69 a2 08 4c   2d
```

Listing 2. »VIZA-SQUEEZER« (Schluß)

```
Name : viza-edit          5dbc 66e9
-----------------------------------
5dbc : 53 ff 00 00 a5 50 48 a5   58
5dc4 : 51 48 20 6e 10 20 52 23   a1
5dcc : 20 5d 61 20 67 61 a9 92   45
5dd4 : 20 d2 ff 20 e4 ff f0 fb   6b
5ddc : c9 85 d0 03 4c 98 61 c9   9f
5de4 : 86 d0 03 4c 0e 62 c9 88   49
5dec : d0 03 4c 17 5e c9 87 d0   28
5df4 : 03 4c 02 61 c9 8c d0 db   c6
5dfc : a9 00 85 c6 68 85 51 68   a8
5e04 : 85 50 20 52 23 20 52 23   c6
5e0c : 20 62 61 a2 03 20 49 08   70
5e14 : 4c 3a 08 a2 07 a9 2e 9d   85
5e1c : 9e 06 9d c6 06 9d ee 06   13
5e24 : 9d 16 07 9d 3e 07 9d 66   a1
5e2c : 07 9d 8e 07 9d b6 07 ca   c8
5e34 : 10 e5 a2 13 bd 2d 66 9d   5c
5e3c : 5a 06 bd 55 66 9d aa 06   bd
5e44 : bd 69 66 9d d2 06 bd 91   7b
5e4c : 66 9d 22 07 bd 41 66 9d   a5
5e54 : 82 06 bd 7d 66 9d fa 06   44
5e5c : ca 10 d9 a9 00 8d 9d 5f   7b
5e64 : 8d 9f 5f 8d a0 5f a2 0d   f4
5e6c : bd a5 66 9d 77 07 bd b3   57
5e74 : 66 9d 9f 07 ca 10 f1 a9   ba
5e7c : 80 8d 8a 02 20 a4 5f 20   8b
5e84 : c3 5f 20 fe 5f 85 c6 ad   77
5e8c : 9d 5f 8d af 07 a9 00 85   fb
5e94 : c6 20 17 60 ad a2 5f d0   4b
5e9c : 05 a9 00 8d a1 5f 20 e4   87
5ea4 : ff f0 fb ae a1 5f f0 03   cf
5eac : 2d a2 5f c9 85 d0 0e a0   94
5eb4 : 07 b1 50 49 ff 91 50 88   b0
5ebc : 10 f7 4c 80 5e c9 86 d0   db
5ec4 : 06 ee 9d 5f 4c 80 5e c9   6a
5ecc : 87 d0 06 ce 9d 5f 4c 80   1e
5ed4 : 5e c9 8c d0 03 4c c6 5d   bc
5edc : c9 11 d0 0d 8d a1 5f ae   c4
```

```
5ee4 : a0 5f e0 07 b0 03 ee a0  6d
5eec : 5f c9 91 d0 0b 8d a1 5f  11
5ef4 : ae a0 5f f0 03 ce a0 5f  d0
5efc : c9 1d d0 0d 8d a1 5f ae  ea
5f04 : 9f 5f e0 07 f0 03 ee 9f  8e
5f0c : 5f c9 9d d0 0b 8d a1 5f  34
5f14 : ae 9f 5f f0 03 ce 9f 5f  6c
5f1c : c9 93 d0 0c a0 07 a9 00  4d
5f24 : 91 50 88 10 fb 4c 80 5e  e2
5f2c : c9 20 d0 1a ac a0 5f a9  1d
5f34 : 80 ae 9f 5f f0 04 4a ca  cd
5f3c : d0 fc 49 ff 31 50 91 50  59
5f44 : ad a1 5f 4c a7 5e c9 2a  0c
5f4c : d0 1a ac a0 5f a9 80 ae  0b
5f54 : 9f 5f f0 04 4a ca d0 fc  98
5f5c : 11 50 91 50 ad a1 5f 4c  02
5f64 : a7 5e d0 00 c9 88 d0 03  99
5f6c : 4c 52 60 c9 89 d0 26 ad  46
5f74 : a2 5f 49 ff 8d a2 5f f0  65
5f7c : 0e a2 02 bd d0 66 9d 90  eb
5f84 : 06 ca 10 f7 4c 80 5e a2  7a
5f8c : 02 bd d3 66 9d 90 06 ca  3a
5f94 : 10 f7 4c 80 5e 4c 80 5e  ca
5f9c : 60 00 00 00 00 00 00 60  bd
5fa4 : a9 00 8d 9e 5f ad 9d 5f  1d
5fac : 0a 2e 9e 5f 0a 2e 9e 5f  ac
5fb4 : 0a 2e 9e 5f 85 50 18 ad  ff
5fbc : 9e 5f 69 38 85 51 60 a9  23
5fc4 : 9e 8d df 5f a9 06 8d e0  cf
5fcc : 5f a0 00 b1 50 48 a2 00  83
5fd4 : 68 0a 48 90 03 a9 2a 2c  e4
5fdc : a9 2e 9d 00 04 e8 e0 08  1f
5fe4 : d0 ee 68 18 ad df 5f 69  73
5fec : 28 8d df 5f ad e0 5f 69  f1
5ff4 : 00 8d e0 5f c8 c0 08 d0  33
5ffc : d2 60 18 a2 16 a0 17 20  5c
6004 : f0 ff a9 60 8d 88 07 8d  bf
600c : 89 07 a9 00 ae 9d 5f 4c  71
6014 : cd bd 60 a9 9e 8d 4a 60  4d
601c : 8d 4f 60 a9 06 8d 4b 60  59
6024 : 8d 50 60 ae a0 5f f0 1a  c4
602c : 18 ad 4a 60 69 28 8d 4a  5c
6034 : 60 8d 4f 60 ad 4b 60 69  c4
603c : 00 8d 4b 60 8d 50 60 ca  54
6044 : d0 e6 ae 9f 5f bd 00 04  13
604c : 49 80 9d 00 04 60 ee 20  7c
6054 : d0 a9 00 8d f9 60 a2 08  e8
605c : bd d6 66 9d 04 07 ca 10  95
6064 : f7 ad 9d 5f 8d af 07 20  38
606c : fe 5f ad f9 60 d0 06 20  a9
6074 : a4 5f 20 c3 5f 20 e4 ff  d3
607c : f0 fb c9 85 d0 0b ad 9d  e4
6084 : 5f 49 80 8d 9d 5f 4c 65  2a
608c : 60 c9 03 f0 56 c9 86 d0  1f
6094 : 03 ee 9d 5f c9 87 d0 03  84
609c : ce 9d 5f c9 0d d0 c2 ad  08
60a4 : f9 60 d0 18 ad 9d 5f 8d  65
60ac : f8 60 a9 01 8d f9 60 a2  ce
60b4 : 09 bd df 66 9d 04 07 ca  0c
60bc : 10 f7 30 a5 ae 9d 5f ad  39
60c4 : f8 60 8d 9d 5f 20 a4 5f  4c
60cc : a0 07 b1 50 99 fa 60 88  6a
60d4 : 10 f8 8a 8d 9d 5f 20 a4  53
60dc : 5f a0 07 b9 fa 60 91 50  1e
60e4 : 88 10 f8 ce 20 d0 a2 13  c5
60ec : bd 7d 66 9d fa 06 ca 10  e0
60f4 : f7 4c 80 5e 00 00 00 00  fd
60fc : 00 00 00 00 00 00 18 a2  a2
6104 : 10 a0 04 20 f0 ff a9 c1  a3
610c : a0 66 20 1e ab 18 a2 10  d1
6114 : a0 13 20 f0 ff a2 00 20  b9
611c : cf ff c9 0d f0 08 9d 58  75
6124 : 63 e8 e0 10 d0 f1 8a a2  41
612c : 56 a0 63 20 bd ff a9 08  42
6134 : aa a0 6f 20 ba ff 20 c0  bc
613c : ff a9 08 20 c3 ff a9 08  09
6144 : aa a8 20 ba ff a9 00 85  fa
614c : 50 a9 38 85 51 a9 50 a2  18
6154 : 00 a0 40 20 d8 ff 4c c6  05
615c : 5d a9 37 85 01 60 a9 30  26
6164 : 85 01 60 a9 1c a0 65 20  54
616c : 1e ab a9 e5 a0 65 20 1e  79
6174 : ab 18 a2 00 a0 18 20 f0  01
617c : ff a2 00 bd 58 63 f0 09  fb
6184 : 20 d2 ff e8 e0 10 d0 f3  e4
618c : 60 a9 20 20 d2 ff e8 e0  5f
6194 : 10 d0 f8 60 18 a2 10 a0  6f
619c : 07 20 f0 ff 8e 0d 62 20  0a
61a4 : fc 64 20 cf ff a5 90 d0  e5
61ac : 4c 20 cf ff a5 90 d0 45  a9
61b4 : 20 cf ff a6 90 d0 3e aa  6e
61bc : 20 cf ff a4 90 d0 36 20  01
61c4 : cd bd 20 cf ff a6 90 d0  8b
61cc : 2c c9 20 d0 02 a9 a0 20  2f
61d4 : d2 ff d0 ee a9 0d 20 d2  e1
61dc : ff ee 0d 62 ad 0d 62 c9  42
61e4 : 17 d0 06 ce 0d 62 20 4a  b8
61ec : 64 18 ae 0d 62 a0 07 20  31
61f4 : f0 ff 4c a6 61 20 cc ff  16
61fc : a9 7e 20 c3 ff a9 00 85  bd
6204 : c6 a5 c6 f0 fc 4c c6 5d  74
620c : 24 00 20 fc 64 a9 00 8d  87
6214 : 09 64 20 e3 63 20 0c 64  04
621c : 18 a2 10 a0 08 20 f0 ff  e3
6224 : 8e 0d 62 a2 04 20 cf ff  a6
622c : a5 90 d0 55 ca d0 f6 20  47
6234 : cf ff a6 90 d0 4b 20 d2  4c
623c : ff d0 f4 a9 0d 20 d2 ff  33
6244 : a2 04 20 cf ff a5 90 d0  fb
624c : 38 ca d0 f6 20 cf ff a6  ca
6254 : 90 d0 2e c9 22 f0 02 d0  64
625c : f3 a0 00 20 cf ff a6 90  5c
6264 : d0 1f c9 22 f0 07 91 50  a9
626c : c8 c0 10 d0 ee ee 09 64  06
6274 : 20 e3 63 20 cf ff a8 a6  50
627c : 90 d0 06 98 d0 f5 4c 44  7f
6284 : 62 20 cc ff a9 7e 20 c3  c0
628c : ff ce 09 64 ad 09 64 8d  91
6294 : 0b 64 c9 ff d0 03 4c 52  3f
629c : 63 a9 ff 8d 09 64 a2 06  d0
62a4 : ee 09 64 ee 0d 62 8a 48  ac
62ac : 18 ae 0d 62 a0 0c 20 f0  78
62b4 : ff 68 aa 20 e3 63 a0 00  72
62bc : b1 50 20 d2 ff c8 c0 10  61
62c4 : d0 f6 a9 0d 20 d2 ff ad  0f
62cc : 09 64 cd 0b 64 f0 03 ca  4c
62d4 : d0 ce a9 00 8d 09 64 a2  6e
62dc : 11 8e 0d 62 20 2e 64 a9  1c
62e4 : 00 85 c6 20 e4 ff f0 fb  66
62ec : c9 03 d0 03 4c 52 63 c9  44
62f4 : 91 d0 03 4c 68 63 c9 11  23
62fc : d0 03 4c 92 63 c9 0d d0  0e
6304 : de ad 09 64 20 e3 63 a0  78
630c : 00 b1 50 99 58 63 f0 05  9b
6314 : c8 c0 10 d0 f4 98 a2 58  aa
631c : a0 63 20 bd ff a9 08 aa  f0
6324 : a0 62 20 ba ff 20 c0 ff  59
632c : a2 08 20 c6 ff 20 cf ff  f3
6334 : d0 14 20 cf ff c9 38 d0  e1
633c : 0d 20 cc ff a9 08 20 c3  6f
6344 : ff a9 00 20 d5 ff 20 cc  93
634c : ff a9 08 20 c3 ff 4c c6  21
6354 : 5d 00 53 3a 20 20 20 20  91
635c : 20 20 20 20 20 20 20 20  5c
6364 : 20 20 20 20 ad 0d 62 c9  01
636c : 11 f0 0c 20 2e 64 ce 09  50
6374 : 64 ce 0d 62 4c e0 62 ad  80
637c : 09 64 f0 0f ce 09 64 20  dc
6384 : 2e 64 20 b2 64 20 c7 63  70
638c : 4c e0 62 4c e3 62 ad 0d  8d
6394 : 62 c9 16 f0 14 ad 09 64  1a
639c : cd 0b 64 f0 23 ee 09 64  bd
63a4 : 20 2e 64 ee 0d 62 4c e0  a9
63ac : 62 ad 09 64 cd 0b 64 f0  5c
63b4 : 0f ee 09 64 20 2e 64 20  4f
63bc : 4a 64 20 c7 63 4c e0 62  1a
63c4 : 4c e3 62 ad 09 64 20 e3  4c
63cc : 63 18 ae 0d 62 a0 0c 20  24
63d4 : f0 ff a0 00 b1 50 20 d2  b0
63dc : ff c8 c0 10 d0 f6 60 a9  0b
63e4 : 00 8d 0a 64 ad 09 64 18  9f
63ec : 0a 2e 0a 64 0a 2e 0a 64  20
63f4 : 0a 2e 0a 64 0a 2e 0a 64  28
63fc : 18 69 00 85 50 a9 6c 6d  58
6404 : 0a 64 85 51 60 00 00 00  d2
640c : a9 00 aa 9d 00 6c 9d 00  ed
6414 : 6d 9d 00 6e 9d 00 6f 9d  f1
641c : 00 70 9d 00 71 9d 00 72  a5
6424 : 9d 00 73 9d 00 74 ca d0  c2
642c : e2 60 ae 0d 62 20 f0 e9  4a
6434 : 20 24 ea a0 0a b1 d1 49  3d
643c : 80 91 d1 ad 86 02 91 f3  55
6444 : c8 c0 1e d0 f0 60 a5 ac  10
644c : 48 a5 ad 48 a5 ae 48 a5  17
6454 : af 48 a2 10 e8 20 f0 e9  f9
645c : e0 16 b0 0c bd f1 ec 85  1f
6464 : ac b5 da 20 c8 e9 30 ec  1c
646c : 20 9d 64 ad 8d 02 c9 02  3e
6474 : f0 10 c9 04 d0 16 a2 00  a7
647c : a8 ea 88 d0 fc ca d0 f9  33
6484 : f0 0a ad 8d 02 d0 fb ad  88
648c : 8d 02 f0 fb 68 85 af 68  18
6494 : 85 ae 68 85 ad 68 85 ac  c9
649c : 60 a0 27 20 f0 e9 20 24  41
64a4 : ea a9 60 91 d1 ad 86 02  56
64ac : 91 f3 88 10 f4 60 a5 ac  9d
64b4 : 48 a5 ad 48 a5 ae 48 a5  7f
64bc : af 48 a2 16 ca 20 f0 e9  40
64c4 : e0 10 f0 0c bd f1 ec 85  94
64cc : ac b5 da 20 e5 64 30 ec  2a
64d4 : e8 20 9d 64 68 85 af 68  02
64dc : 85 ae 68 85 ad 68 85 ac  11
64e4 : 60 29 03 09 04 85 ad 20  1e
64ec : e0 e9 a0 27 b1 d1 91 ac  17
64f4 : b1 f3 91 ae 88 10 f5 60  7b
64fc : a9 01 a2 0c a0 62 20 bd  69
6504 : ff a9 7e a2 08 a0 00 20  92
650c : ba ff 20 c0 ff a2 7e 20  35
6514 : c6 ff 20 cf ff 4c cf ff  7d
651c : 13 12 d6 49 5a 41 2d c5  07
6524 : 44 49 54 4f 52 20 20 20  f3
652c : 20 20 20 ce 41 4d 45 20  12
6534 : 3a 20 20 20 20 20 20 20  4e
653c : 20 20 20 20 20 20 20 20  3c
6544 : 20 20 92 11 1d 1d c0 c0  7a
654c : c0 c0 c0 c0 c0 c0 c0 c0  4b
6554 : c0 c0 c0 c0 c0 c0 c0 c0  53
655c : c0 c0 c0 c0 c0 c0 c0 c0  5b
6564 : c0 c0 c0 c0 c0 c0 c0 c0  63
656c : c0 c0 0d 1d 1d 1d 1d 1d  dd
6574 : 1d 1d 46 31 a0 2d a0 c4  57
657c : 49 53 50 4c 41 59 20 c4  f5
6584 : 49 52 45 43 54 4f 52 59  6c
658c : 0d 1d 1d 1d 1d 1d 1d 1d  7c
6594 : 46 33 a0 2d a0 cc 4f 41  72
659c : 44 20 da 45 49 43 48 45  aa
65a4 : 4e 53 41 54 5a 0d 1d 1d  33
65ac : 1d 1d 1d 1d 1d 46 35 a0  5d
65b4 : 2d a0 d3 41 56 45 20 da  14
65bc : 45 49 43 48 45 4e 53 41  16
65c4 : 54 5a 0d 1d 1d 1d 1d 1d  96
65cc : 1d 1d 46 37 a0 2d a0 c5  72
65d4 : 44 49 54 20 da 45 49 43  59
65dc : 48 45 4e 53 41 54 5a 0d  ff
65e4 : 00 1d 1d 1d 1d 1d 1d 1d  c7
65ec : 46 38 a0 2d a0 d2 45 54  7a
65f4 : 55 52 4e 20 54 4f 20 d6  f8
65fc : 49 5a 41 57 52 49 54 45  f9
6604 : 0d 1d 1d c0 c0 c0 c0 c0  96
660c : c0 c0 c0 c0 c0 c0 c0 c0  0b
6614 : c0 c0 c0 c0 c0 c0 c0 c0  13
661c : c0 c0 c0 c0 c0 c0 c0 c0  1b
6624 : c0 c0 c0 c0 c0 c0 c0 12  c6
662c : 00 46 31 60 2d 60 49 0e  bf
6634 : 16 05 12 14 09 05 12 05  df
663c : 0e 60 60 60 60 46 32 60  60
6644 : 2d 60 41 15 14 0f 0d 0f  a0
664c : 16 05 20 01 15 13 60 60  39
6654 : 60 46 33 60 2d 60 5a 05  fa
665c : 09 03 08 05 0e 03 0f 04  c7
6664 : 05 2b 60 60 60 46 35 60  f1
666c : 2d 60 5a 05 09 03 08 05  d3
6674 : 0e 03 0f 04 05 2d 60 60  44
667c : 60 46 37 60 2d 60 43 0f  da
6684 : 10 19 20 43 08 01 12 01  64
668c : 03 14 05 12 60 46 38 60  f7
6694 : 2d 60 52 05 14 15 12 0e  75
669c : 20 14 0f 20 4d 05 0e 15  ee
66a4 : 60 43 08 01 12 01 03 14  25
66ac : 05 12 60 4e 12 2e 3a 43  9e
66b4 : 08 01 12 01 03 14 05 12  eb
66bc : 60 3a 60 60 60 d0 52 4f  d2
66c4 : 47 52 41 4d 4d 4e 41 4d  15
66cc : 45 20 3a 00 01 0e 60 01  b4
66d4 : 15 13 17 08 01 14 20 3f  e9
66dc : 60 60 60 14 0f 20 17 08  65
66e4 : 05 12 05 20 3f b7 62 f0  55
```

Listing 3. »VIZA-EDIT«. Zum Verändern des Bildschirm-Zeichensatzes von Vizawrite. Bitte mit dem MSE (Seite 159) eingeben.

```
Name : viza-speller         5dbc 6e75
-------------------------------------
5dbc : 53 ff 00 00 a2 0a b5 50  01
5dc4 : 9d 49 6e ca 10 f8 20 5f  03
5dcc : 5e 20 5f 6c a9 00 85 c6  de
5dd4 : 20 e4 ff a2 0b dd 1a 5e  7f
5ddc : f0 05 ca 10 f8 30 f1 8a  f1
5de4 : 0a aa bd 26 5e 8d 04 5e  97
5dec : bd 27 5e 8d 05 5e 8e 45  8e
5df4 : 6e a0 03 bd 28 04 09 80  f3
5dfc : 9d 28 04 e8 88 d0 f4 20  ef
5e04 : ff ff ae 45 6e a0 03 bd  cb
5e0c : 28 04 29 7f 9d 28 04 e8  6e
5e14 : 88 d0 f4 4c cd 5d 43 44  28
```

Listing 4. »VIZA-SPELLER«. Durch diesen sehr einfach zu bedienenden Spell-Checker lehren Sie Ihren Tippfehler-Teufel das Fürchten. Bitte mit dem MSE (Seite 159) eingeben.

```
5e1c : 45 46 49 54 55 57 c3 cd  1c
5e24 : d0 d6 fd 64 11 6a e9 68  48
5e2c : 16 68 08 63 77 5e 69 64  be
5e34 : d5 62 c8 63 64 64 80 64  0d
5e3c : 3e 5e a2 0a bd 49 6e 95  9e
5e44 : 50 ca 10 f8 68 68 a9 00  8d
5e4c : 85 c6 20 52 23 20 52 23  49
5e54 : 20 4c 08 a2 03 20 49 08  57
5e5c : 4c 3a 08 20 4c 08 20 3a  c5
5e64 : 08 20 59 6c a9 80 8d 8a  4a
5e6c : 02 a9 96 a0 6c 20 76 69  71
5e74 : 4c 0e 62 a9 52 a0 6d 20  b5
5e7c : 76 69 20 53 6c ad 44 0b  75
5e84 : 85 52 8d e4 6d ad 45 0b  a2
5e8c : 85 53 8d e5 6d a9 ff a0  40
5e94 : 00 91 52 a5 2b 85 50 a5  12
5e9c : 2c 85 51 a9 00 8d ec 6d  0f
5ea4 : a9 00 8d ed 6d a9 00 8d  ae
5eac : ee 6d a9 00 8d ef 6d 20  09
5eb4 : dd 5f b0 1a ee ec 6d d0  5e
5ebc : 03 ee ed 6d 20 33 60 90  9e
5ec4 : ee ee ee 6d d0 03 ee ef  53
5ecc : 6d 20 1c 61 90 e1 20 a4  5e
5ed4 : 69 20 59 6c a2 03 4c a6  f2
5edc : 69 00 01 02 03 04 05 06  36
5ee4 : 07 08 09 0a 0b 0c 0d 0e  d4
5eec : 0f 10 11 12 13 14 15 16  dc
5ef4 : 17 18 19 1a 00 00 00 00  a1
5efc : 00 00 00 00 00 00 00 00  fd
5f04 : 00 00 00 00 00 00 00 00  05
5f0c : 00 00 00 00 00 00 00 00  0d
5f14 : 00 00 00 00 00 00 00 00  15
5f1c : 00 00 01 02 03 04 05 06  0d
5f24 : 07 08 09 0a 0b 0c 0d 0e  14
5f2c : 0f 10 11 12 13 14 15 16  1c
5f34 : 17 18 19 1a 00 00 00 00  e1
5f3c : 00 00 00 00 00 00 1b 00  a9
5f44 : 00 00 00 00 00 00 00 00  45
5f4c : 00 00 00 00 00 00 00 1c  85
5f54 : 00 1d 1b 1c 1d 1e 00 00  f0
5f5c : 00 00 00 00 00 00 00 00  5d
5f64 : 00 00 00 00 00 00 00 00  65
5f6c : 00 00 00 00 00 00 00 00  6d
5f74 : 00 00 00 00 00 00 00 00  75
5f7c : 00 00 00 00 00 00 00 00  7d
5f84 : 00 00 00 00 00 00 00 00  85
5f8c : 00 00 00 00 00 00 00 00  8d
5f94 : 00 00 00 00 00 00 00 00  95
5f9c : 00 00 00 00 00 00 00 00  9d
5fa4 : 00 00 00 00 00 00 00 00  a5
5fac : 00 00 00 00 00 00 00 00  ad
5fb4 : 00 00 00 00 00 00 00 00  b5
5fbc : 00 00 00 00 00 00 00 00  bd
5fc4 : 00 00 00 00 00 00 00 00  c5
5fcc : 00 00 00 00 00 00 00 00  cd
5fd4 : 00 00 00 00 00 00 00 00  d5
5fdc : ff a0 00 b1 50 e6 50 d0  81
5fe4 : 02 e6 51 a8 b9 dd 5e f0  a9
5fec : f0 30 42 a2 00 9d 00 70  a7
5ff4 : e8 a0 00 b1 50 e6 50 d0  82
5ffc : 02 e6 51 a8 b9 dd 5e 30  3f
6004 : 2c d0 ea e0 02 90 d2 8e  7c
600c : f0 6d a2 00 bd 00 70 20  39
6014 : 7e 62 9d 78 04 e8 e0 28  95
601c : f0 11 ec f0 6d d0 ed a9  56
6024 : 60 e0 28 f0 06 9d 78 04  54
602c : e8 d0 f6 18 60 38 60 ad  e2
6034 : 44 0b 85 52 ad 45 0b 85  e6
603c : 53 20 86 60 20 d8 61 a5  e7
6044 : 52 8d e6 6d a5 53 8d e7  bf
604c : 6d 20 a8 60 b0 11 20 e9  e7
6054 : 60 f0 0e b0 ea ad e6 6d  58
605c : 85 52 ad e7 6d 85 53 38  33
6064 : 60 ee e2 6d d0 03 ee e3  4a
606c : 6d ad e6 6d 85 52 ad e7  89
6074 : 6d 85 53 a0 02 ad e2 6d  81
607c : 91 52 c8 ad e3 6d 91 52  b3
6084 : 18 60 ac f0 6d b9 ff 6f  99
608c : aa bd 56 6e 99 7f 70 88  e1
6094 : d0 f3 60 ac f1 6d b9 ff  7d
609c : 70 aa bd 56 6e 99 7f 71  30
60a4 : 88 d0 f3 60 a0 00 b1 52  13
60ac : 8d 43 6e 30 2d c8 b1 52  01
60b4 : 8d f1 6d c8 b1 52 8d e2  58
60bc : 6d c8 b1 52 8d e3 6d c8  83
60c4 : a2 00 b1 52 9d 00 71 e8  8e
60cc : c8 ec f1 6d d0 f4 18 98  7b
60d4 : 65 52 85 52 90 02 e6 53  69
60dc : 18 60 38 60 20 53 6c 20  cd
60e4 : a8 60 4c 59 6c 20 97 60  e1
60ec : ad f0 6d cd f1 6d 90 03  f9
60f4 : ad f1 6d 8d e8 6d a2 00  2b
60fc : bd 80 70 dd 80 71 d0 13  ce
6104 : e8 ec e8 6d d0 f2 ad f0  87
610c : 6d cd f1 6d f0 05 b0 04  8c
6114 : 18 a9 01 60 38 a9 01 60  e3
611c : 18 ad f0 6d 69 04 18 6d  e7
6124 : e4 6d 8d e9 6d ad e5 6d  16
612c : 69 00 8d ea 6d 38 ad e9  79
6134 : 6d e9 30 ad ea 6d e9 fe  17
613c : b0 28 20 68 61 a0 00 98  62
6144 : 91 52 c8 ad f0 6d 91 52  4b
614c : 98 c8 91 52 c8 a9 00 91  f4
6154 : 52 c8 a2 00 bd 00 70 91  74
615c : 52 e8 c8 ec f0 6d d0 f4  9a
6164 : 18 60 38 60 ad e4 6d 85  89
616c : 58 ad e5 6d 85 59 ad e9  6f
6174 : 6d 85 5a ad ea 6d 85 5b  d7
617c : e6 58 d0 02 e6 59 e6 5a  8c
6184 : d0 02 e6 5b 38 a5 58 e5  58
618c : 52 a8 a5 59 e5 53 aa 8c  84
6194 : e0 6d 38 a5 58 ed e0 6d  41
619c : 85 58 b0 02 c6 59 38 a5  1d
61a4 : 5a ed e0 6d 85 5a b0 02  cd
61ac : c6 5b 98 f0 08 88 b1 58  a0
61b4 : 91 5a 98 d0 f8 8a f0 0f  78
61bc : c6 59 c6 5b 88 b1 58 91  e6
61c4 : 5a 98 d0 f8 ca d0 f1 ad  14
61cc : e9 6d 8d e4 6d ad ea 6d  36
61d4 : 8d e5 6d 60 a0 00 b1 52  31
61dc : 30 2f c8 b1 52 aa a0 04  11
61e4 : b1 52 a8 b9 56 6e cd 80  31
61ec : 70 f0 11 b0 1c 18 8a 69  ae
61f4 : 04 18 65 52 85 52 90 dc  8f
61fc : e6 53 d0 d8 a0 05 b1 52  79
6204 : a8 b9 56 6e cd 81 70 90  b8
620c : e4 60 a2 7f a9 00 9d 00  ca
6214 : 72 ca 10 fa 60 a2 00 bd  e5
621c : 00 70 9d 00 72 e8 ec f0  c0
6224 : 6d d0 f4 8e 46 6e 60 a2  a7
622c : 00 bd 00 71 9d 00 72 e8  af
6234 : ec f1 6d d0 f4 8e 46 6e  48
623c : 60 a2 08 20 c9 ff a2 00  1b
6244 : bd 00 71 dd 00 72 d0 0b  06
624c : e8 ec 46 6e f0 05 ec f1  d8
6254 : 6d d0 ed 8a 8e 47 6e 20  13
625c : a8 ff ad f1 6d 20 a8 ff  28
6264 : ae 47 6e ec f1 6d f0 0c  55
626c : bd 00 71 20 a8 ff ee 47  5e
6274 : 6e 4c 64 62 20 2b 62 4c  eb
627c : cc ff 8e e0 6d aa bd 89  3e
6284 : 62 ae e0 6d 60 00 01 02  31
628c : 03 04 05 06 07 08 09 0a  7c
6294 : 0b 0c 0d 0e 0f 10 11 12  84
629c : 13 14 15 16 17 18 19 1a  8c
62a4 : 65 76 78 7c c9 20 d0 02  d7
62ac : a9 a0 4c d2 ff 8e e0 6d  e5
62b4 : a2 00 dd cc 62 f0 08 e8  07
62bc : bc cc 62 d0 f5 f0 03 bd  ff
62c4 : d1 62 ae e0 6d 29 1f 60  eb
62cc : 65 76 78 7c 00 1b 1c 1d  9e
62d4 : 1e ad 44 0b 85 52 ad 45  67
62dc : 0b 85 53 a2 01 8e 42 6e  3d
62e4 : ca 8e 48 6e 20 2f 63 b0  40
62ec : 0a 20 ef 69 c9 03 f0 06  b4
62f4 : 4c e8 62 20 ef 69 4c fd  c8
62fc : 62 a2 06 20 a6 69 e8 e0  50
6304 : 17 d0 f8 60 a9 69 a0 6d  11
630c : 20 76 69 ad ed 6d ae ec  56
6314 : 6d 20 cd bd a9 83 a0 6d  d0
631c : 20 76 69 ad ef 6d ae ee  8a
6324 : 6d 20 cd bd 20 ef 69 4c  8c
632c : fd 62 60 20 fd 62 a9 06  1c
6334 : 8d eb 6d a9 00 8d f2 6d  5a
633c : a9 13 8d f3 6d 20 e0 60  6d
6344 : b0 e8 ad 48 6e f0 05 ad  bb
634c : 43 6e f0 f1 ac f2 6d ae  b6
6354 : eb 6d 20 0b 6b ad e3 6d  ee
635c : ae e2 6d 20 cd bd ae eb  38
6364 : 6d bd bd 69 85 d1 bd d6  d8
636c : 69 85 d2 18 ad f2 6d 69  4a
6374 : 05 a8 a2 00 bd 00 71 20  58
637c : 7e 62 91 d1 e8 c8 cc f3  ba
6384 : 6d f0 05 ec f1 6d d0 ec  f0
638c : ad 42 6e f0 12 ad 43 6e  8c
6394 : f0 0d 18 ad f2 6d 69 05  11
639c : a8 ae eb 6d 20 f2 65 ee  51
63a4 : eb 6d ad eb 6d c9 17 d0  52
63ac : 94 ad f2 6d c9 14 f0 12  a6
63b4 : a9 06 8d eb 6d a9 14 8d  d1
63bc : f2 6d a9 27 8d f3 6d 4c  7b
63c4 : 41 63 18 60 a9 00 8d 48  2a
63cc : 6e 20 b2 65 20 d4 65 20  22
63d4 : 96 65 b0 1b 20 15 64 b0  4a
63dc : 16 ad 00 70 30 1a 20 fa  21
63e4 : 66 b0 15 20 19 62 20 e9  e5
63ec : 60 90 ee d0 0b f0 e0 20  66
63f4 : 5c 64 20 fd 62 4c a4 69  38
63fc : 20 ec 65 20 b3 67 20 96  14
6404 : 65 b0 ec 20 15 64 b0 e7  07
640c : ad 00 70 30 eb f0 ca 10  6d
6414 : d5 ad f6 6d f0 13 ad 00  89
641c : 71 20 9a 66 cd f7 6d f0  45
6424 : 35 20 5c 64 a9 00 8d 00  de
642c : 70 a9 01 8d f6 6d 20 0e  da
6434 : 62 ad 00 71 20 9a 66 8d  27
643c : f7 6d 8d 03 6e a9 0c a2  57
6444 : f8 a0 6d 20 bd ff a0 02  4e
644c : a9 01 20 4e 6c 20 c0 ff  12
6454 : 20 ba 6a 90 01 60 18 60  b2
645c : 20 cc ff a9 01 4c c3 ff  99
6464 : a9 ff 4c ff 64 ad 44 0b  fb
646c : 85 52 ad 45 0b 85 53 a9  ac
6474 : 01 8d 42 6e a9 ff 8d 48  fc
647c : 6e 4c e8 62 a9 01 a2 55  6f
6484 : a0 6e 20 bd ff a9 04 a2  be
648c : 04 a0 07 20 ba ff 20 c0  54
6494 : ff b0 38 a2 04 20 c9 ff  b6
649c : a6 90 d0 2f ad 44 0b 85  d9
64a4 : 52 ad 45 0b 85 53 20 e0  b5
64ac : 60 b0 20 ad 43 6e f0 f6  7b
64b4 : a2 00 bd 00 71 20 d7 64  06
64bc : e8 ec f1 6d d0 f4 a9 0d  ba
64c4 : 20 d2 ff a9 00 20 d2 ff  cf
64cc : 4c aa 64 20 cc ff a9 04  06
64d4 : 4c c3 ff a8 b9 de 64 4c  d3
64dc : d2 ff 40 41 42 43 44 45  c0
64e4 : 46 47 48 49 4a 4b 4c 4d  d4
64ec : 4e 4f 50 51 52 53 54 55  dc
64f4 : 56 57 58 59 5a 7b 7c 7d  a6
64fc : 7e a9 00 8d 48 6e 20 b2  de
6504 : 65 20 d4 65 20 96 65 b0  09
650c : 1c 20 2d 66 b0 1f ad 00  0b
6514 : 70 30 26 20 fa 66 b0 21  12
651c : 20 e9 60 90 59 d0 1a 20  20
6524 : 4c 67 4c 08 65 20 9a 67  c9
652c : a2 03 20 a6 69 20 b4 66  64
6534 : 20 d0 66 20 fd 62 4c a4  c7
653c : 69 a9 9d a0 6d 20 76 69  7a
6544 : 20 80 65 20 f6 69 c9 03  ea
654c : f0 db c9 4e f0 22 c9 4a  42
6554 : d0 06 20 e5 67 20 3d 62  1d
655c : a2 03 20 a6 69 20 96 65  19
6564 : b0 c3 20 2d 66 b0 c6 ad  06
656c : 00 70 30 cd f0 a0 10 a8  10
6574 : 20 b3 67 4c 5c 65 20 4c  db
657c : 67 4c 12 65 a2 00 bd 00  5c
6584 : 71 20 7e 62 9d 78 04 e8  71
658c : e0 28 f0 05 ec f1 6d d0  13
6594 : ed 60 20 ec 65 20 09 66  9f
659c : 20 e0 60 b0 0b ad 48 6e  77
65a4 : f0 05 ad 43 6e f0 f1 18  51
65ac : 08 20 ec 65 28 60 ad 44  71
65b4 : 0b 85 52 ad 45 0b 85 53  36
65bc : a9 05 8d f4 6d a9 05 8d  3d
65c4 : f5 6d a9 00 8d f6 6d 8d  3c
65cc : 42 6e 8d 00 70 4c ec 65  91
65d4 : a5 52 8d e6 6d a5 53 8d  4f
65dc : e7 6d 20 2f 63 ad e6 6d  82
65e4 : 85 52 ad e7 6d 85 53 60  0c
65ec : ae f4 6d ac f5 6d bd bd  42
65f4 : 69 85 d1 bd d6 69 85 d2  c0
65fc : a2 0d b1 d1 49 80 91 d1  4e
6604 : c8 ca 10 f6 60 ee f4 6d  40
660c : ad f4 6d c9 17 d0 11 a9  57
6614 : 06 8d f4 6d ad f5 6d c9  9f
661c : 19 f0 06 a9 19 8d f5 6d  15
6624 : 60 a9 05 8d f5 6d 4c d4  f1
662c : 65 ad f6 6d f0 1b ad 00  72
6634 : 71 20 9a 66 cd f7 6d f0  5d
663c : 5b 20 9a 67 20 b4 66 20  bc
6644 : d0 66 b0 4f a9 00 8d 00  2e
664c : 70 a9 01 8d f6 6d 20 0e  fa
6654 : 62 ad 00 71 20 9a 66 8d  47
665c : f7 6d 8d 03 6e 8d 30 6e  bf
6664 : 20 d9 66 b0 2e a9 0c a2  c6
666c : f8 a0 6d 20 bd ff a0 01  74
6674 : a9 08 20 4e 6c 20 c0 ff  be
667c : 20 ba 6a b0 16 a9 0f a2  da
6684 : 04 a0 6e 20 bd ff a0 02  da
668c : a9 01 20 4e 6c 20 c0 ff  52
6694 : 4c ba 6a 60 18 60 a2 03  f9
669c : dd ac 66 f0 05 ca 10 f8  60
66a4 : 30 03 bd b0 66 09 40 60  4c
66ac : 1b 1c 1d 1e 01 0f 15 13  e4
66b4 : a2 08 20 c9 ff a9 ff 20  29
66bc : a8 ff 20 cc ff a9 08 20  b3
66c4 : c3 ff a9 01 20 c3 ff a9  85
66cc : 0f 4c c3 ff a9 11 a2 31  02
66d4 : a0 6e 4c df 66 a9 1e a2  2c
66dc : 13 a0 6e 20 bd ff a9 0f  80
66e4 : a2 08 a8 20 ba ff a9 0f  29
66ec : 20 c3 ff 20 c0 ff a9 0f  c2
66f4 : 20 c3 ff 4c ba 6a a2 01  0b
66fc : 20 c6 ff a2 00 8e 47 6e  42
6704 : 8e 44 6e 20 a5 ff 30 38  df
670c : f0 14 8d 46 6e a2 00 bd  aa
6714 : 00 72 9d 00 70 e8 ec 46  43
671c : 6e d0 f4 8e 47 6e 20 a5  b5
6724 : ff 8d f0 6d ae 47 6e ec  8c
672c : f0 6d f0 0c 20 a5 ff 9d  fb
6734 : 00 70 ee 47 6e 4c 28 67  ca
673c : 20 cc ff 20 86 60 18 60  53
6744 : 8d 00 70 20 cc ff 38 60  60
```

```
674c : a9 01 8d 44 6e ad 00 70  97
6754 : 30 43 f0 41 a2 08 20 c9  09
675c : ff a2 00 bd 00 70 dd 00  5f
6764 : 72 d0 0b e8 ec 46 6e f0  bb
676c : 05 ec f0 6d d0 ed 8a 8e  95
6774 : 47 6e 20 a8 ff ad f0 6d  1b
677c : 20 a8 ff ae 47 6e ec f0  43
6784 : 6d f0 0c bd 00 70 20 a8  79
678c : ff ee 47 6e 4c 7f 67 20  41
6794 : cc ff 4c 19 62 60 ad 00  76
679c : 70 30 13 ad 44 6e d0 03  a0
67a4 : 20 4c 67 20 fa 66 b0 06  7a
67ac : 20 4c 67 4c a7 67 60 20  cd
67b4 : 53 6c 38 a5 52 e9 04 85  90
67bc : 52 b0 02 c6 53 38 a5 52  f2
67c4 : ed f1 6d 85 52 b0 02 c6  f6
67cc : 53 a0 00 a9 01 91 52 18  bb
67d4 : ad f1 6d 69 04 18 65 52  3e
67dc : 85 52 90 02 e6 53 4c 59  dc
67e4 : 6c 20 53 6c 38 a5 52 e9  91
67ec : 04 85 52 b0 02 c6 53 38  72
67f4 : a5 52 ed f1 6d 85 52 b0  2a
67fc : 02 c6 53 a0 00 98 91 52  fa
6804 : 18 ad f1 6d 69 04 18 65  ff
680c : 52 85 52 90 02 e6 53 4c  05
6814 : 59 6c a9 27 a0 00 a2 03  8d
681c : 20 11 6b b0 56 f0 54 ae  51
6824 : f6 6b bd 00 02 20 b1 62  ec
682c : 9d 00 71 ca 10 f4 20 0e  c4
6834 : 62 ad 00 71 20 9a 66 8d  27
683c : 03 6e a9 0c a2 f8 a0 6d  b1
6844 : 20 bd ff a0 02 a9 01 20  08
684c : 4e 6c 20 c0 ff 20 ba 6a  b1
6854 : b0 21 20 84 68 20 fa 66  6e
685c : b0 10 20 19 62 20 8a 68  62
6864 : d0 f3 20 a8 68 b0 06 4c  08
686c : 59 68 20 ef 69 20 77 68  46
6874 : 4c 16 68 a9 01 20 c3 ff  3b
687c : 20 fd 62 a2 03 4c a6 69  88
6884 : a9 06 8d eb 6d 60 ad f0  84
688c : 6d cd f6 6b 90 13 a2 00  37
6894 : bd 00 70 dd 00 71 d0 09  0a
689c : e8 ec f6 6b d0 f2 a9 00  71
68a4 : 60 a9 01 60 ae eb 6d bd  a1
68ac : bd 69 85 d1 bd d6 69 85  fd
68b4 : d2 a0 00 a2 00 bd 00 70  fa
68bc : 20 7e 62 91 d1 c8 e8 c0  6f
68c4 : 28 f0 05 ec f0 6d d0 ed  dd
68cc : ee eb 6d ad eb 6d c9 17  40
68d4 : d0 0f a9 06 8d eb 6d 20  85
68dc : ef 69 c9 03 f0 05 20 fd  06
68e4 : 62 18 60 38 60 a9 27 a0  a3
68ec : 00 a2 03 20 11 6b b0 2e  8e
68f4 : f0 2c ae f6 6b bd 00 02  2d
68fc : 20 b1 62 9d 00 71 ca 10  18
6904 : f4 a9 00 8d f6 6d 20 2d  34
690c : 66 20 fa 66 b0 10 20 8a  2f
6914 : 68 f0 06 20 4c 67 4c 0d  c5
691c : 69 20 2d 69 90 eb 20 b4  60
6924 : 66 a2 03 20 a6 69 4c d0  29
692c : 66 a2 03 20 a6 69 a2 00  e8
6934 : bd 00 70 20 7e 62 9d 78  74
693c : 04 e8 e0 28 f0 05 ec f0  be
6944 : 6d d0 ed a9 c3 a0 6d 20  01
694c : 76 69 20 f6 69 c9 03 f0  31
6954 : 11 c9 4a f0 03 20 4c 67  2c
695c : a2 03 20 a6 69 20 19 62  1d
6964 : 18 60 a2 03 20 a6 69 20  d2
696c : 4c 67 20 19 62 20 9a 67  f7
6974 : 38 60 8d 99 69 8c 9a 69  ab
697c : 20 98 69 aa e0 0d f0 1f  10
6984 : 20 98 69 a8 20 0b 6b 20  a8
698c : 98 69 c9 40 f0 ea 20 a8  8b
6994 : 62 4c 8b 69 ad ff ff ee  e5
699c : 99 69 d0 03 ee 9a 69 60  a8
69a4 : a2 18 bd bd 69 8d b7 69  2e
69ac : bd d6 69 8d b8 69 a0 27  88
69b4 : a9 60 99 ff ff 88 10 fa  6e
69bc : 60 00 28 50 78 a0 c8 f0  c2
69c4 : 18 40 68 90 b8 e0 08 30  3c
69cc : 58 80 a8 d0 f8 20 48 70  3b
69d4 : 98 c0 04 04 04 04 04 04  c6
69dc : 04 05 05 05 05 05 05 06  dd
69e4 : 06 06 06 06 06 06 07 07  ea
69ec : 07 07 07 a9 36 a0 6d 20  cc
69f4 : 76 69 a9 00 85 c6 20 68  69
69fc : 6c 20 e4 ff f0 fb 48 20  02
6a04 : a4 69 68 60 a9 00 85 c6  c1
6a0c : a5 c6 f0 fc 60 a9 01 a2  8d
6a14 : df a0 6d 20 bd ff a0 00  01
6a1c : a9 08 20 4e 6c 20 c0 ff  66
6a24 : a2 08 20 c6 ff 20 84 6a  93
6a2c : 20 84 6a a9 06 d0 02 a9  a0
6a34 : 07 8d e1 6d 20 91 6a ae  bd
6a3c : e1 6d a0 07 20 0b 6b 20  25
6a44 : 84 6a 20 84 6a 20 84 6a  25
6a4c : 8d e0 6d 20 84 6a ae e0  c1
6a54 : 6d 20 cd bd 20 84 6a 20  0c
6a5c : a8 62 d0 f8 a9 0d 20 a8  5d
6a64 : 62 ee e1 6d ad e1 6d c9  96
6a6c : 17 d0 cc 20 a8 6a a6 90  bc
6a74 : f0 bd d0 03 20 a8 6a 20  09
6a7c : a0 6a 20 ba 6a 4c fd 62  76
6a84 : 20 a5 ff a6 90 f0 05 68  c1
6a8c : 68 4c 78 6a 60 a2 07 20  fd
6a94 : a6 69 e8 e0 17 d0 f8 60  e2
6a9c : 04 04 00 ff 20 cc ff a9  5e
6aa4 : 08 4c c3 ff 20 cc ff 20  6c
6aac : ef 69 c9 03 d0 03 4c 7b  70
6ab4 : 6a a2 08 4c c6 ff a9 0f  2c
6abc : a2 08 a8 20 ba ff a9 00  e3
6ac4 : 20 bd ff 20 c0 ff a2 0f  7b
6acc : 20 c6 ff a0 00 20 a5 ff  fb
6ad4 : a6 90 d0 06 99 00 02 c8  ea
6adc : d0 f3 a9 00 99 00 02 20  f2
6ae4 : cc ff ad 00 02 0d 01 02  ac
6aec : c9 30 d0 02 18 60 a0 00  49
6af4 : a2 18 20 0b 6b a2 00 bd  53
6afc : 00 02 20 a8 62 f0 03 e8  a6
6b04 : d0 f5 20 f6 69 38 60 18  c0
6b0c : 4c f0 ff 38 60 8d f5 6b  f8
6b14 : 8c f3 6b 84 d3 bd bd 69  fa
6b1c : 85 d1 bd d6 69 85 d2 18  12
6b24 : ad f3 6b 6d f5 6b 8d f4  2e
6b2c : 6b 85 d5 a9 00 85 c6 a5  97
6b34 : c6 85 cc 8d 92 02 f0 f7  8e
6b3c : 78 a5 cf f0 0e a5 ce ae  3f
6b44 : 87 02 a0 00 84 d4 84 cf  95
6b4c : 20 13 ea 20 b4 e5 20 30  10
6b54 : 75 c9 03 f0 b6 c9 14 d0  38
6b5c : 0a a6 d3 ec f3 6b f0 cf  4a
6b64 : 4c 00 6c c9 0d f0 47 c9  0e
6b6c : 1d d0 0b a6 d3 ec f4 6b  d8
6b74 : f0 bd e6 d3 d0 2c c9 20  4d
6b7c : 90 b5 c9 9d d0 09 a6 d3  a4
6b84 : ec f3 6b f0 aa d0 18 c9  88
6b8c : 94 d0 03 4c 2d 6c c9 80  31
6b94 : 90 06 c9 a0 b0 02 90 97  3a
6b9c : a6 d3 ec f4 6b b0 90 20  c4
6ba4 : 16 e7 a5 d3 cd f4 6b d0  65
6bac : 02 c6 d3 4c 33 6b 20 f7  8f
6bb4 : 6b ac f4 6b b1 d1 c9 60  b1
6bbc : d0 0c 88 cc f3 6b d0 f4  16
6bc4 : b1 d1 c9 60 f0 1f c8 8c  20
6bcc : f6 6b a2 00 ac f3 6b b1  9c
6bd4 : d1 20 18 6c 9d 00 02 e8  fd
6bdc : c8 cc f6 6b d0 f1 ad 00  89
6be4 : 02 c9 60 d0 02 a2 00 8e  4f
6bec : f6 6b ad f6 6b 18 60 00  db
6bf4 : 00 00 00 a4 d3 b1 d1 29  ed
6bfc : 7f 91 d1 60 a4 d3 c6 d3  70
6c04 : b1 d1 88 91 d1 c8 c8 cc  12
6c0c : f4 6b d0 f4 88 a9 60 91  03
6c14 : d1 4c 33 6b c9 20 90 09  38
6c1c : c9 40 90 04 c9 60 90 05  96
6c24 : 60 18 69 40 60 18 69 80  60
6c2c : 60 ac f4 6b 88 b1 d1 c9  7e
6c34 : 60 d0 14 c4 d3 f0 10 c8  31
6c3c : 88 88 b1 d1 c8 91 d1 c4  99
6c44 : d3 d0 f5 a9 60 91 d1 4c  a4
6c4c : 33 6b a2 08 4c ba ff 78  6a
6c54 : a0 30 84 01 60 a0 37 84  3f
6c5c : 01 58 60 a9 11 a2 09 a0  62
6c64 : 80 4c 6e 6c a9 15 a2 19  33
6c6c : a0 fc 8d 01 d4 8e 05 d4  8d
6c74 : 8c 06 d4 a9 11 8d 00 d4  95
6c7c : a9 11 8d 04 d4 a9 0f 8d  84
6c84 : 18 d4 a2 a0 a0 00 88 d0  91
6c8c : fd ca d0 f8 a9 10 8d 04  9b
6c94 : d4 60 00 00 12 56 49 5a  46
6c9c : 41 2d 53 50 45 4c 4c 45  c5
6ca4 : 52 20 56 33 2e 30 20 20  28
6cac : 20 20 20 20 20 20 42 59  a7
6cb4 : 20 c4 49 45 54 45 52 20  2a
6cbc : c2 41 59 45 52 92 40 01  db
6cc4 : 00 20 43 20 44 20 45 20  44
6ccc : 46 20 49 20 54 20 55 20  55
6cd4 : 57 20 c3 20 cd 20 d0 20  92
6cdc : d6 20 40 02 00 c0 c0 c0  9d
6ce4 : c0 c0 c0 c0 c0 c0 c0 c0  e3
6cec : c0 c0 c0 c0 c0 c0 c0 c0  eb
6cf4 : c0 c0 c0 c0 c0 c0 c0 c0  f3
6cfc : c0 c0 c0 c0 c0 c0 c0 c0  fb
6d04 : c0 c0 c0 c0 c0 40 04 00  8b
6d0c : c0 c0 c0 c0 c0 c0 c0 c0  0b
6d14 : c0 c0 c0 c0 c0 c0 c0 c0  13
6d1c : c0 c0 c0 c0 c0 c0 c0 c0  1b
6d24 : c0 c0 c0 c0 c0 c0 c0 c0  23
6d2c : c0 c0 c0 c0 c0 c0 c0 c0  2b
6d34 : 40 0d 18 08 c2 49 54 54  72
6d3c : 45 20 45 49 4e 45 20 d4  45
6d44 : 41 53 54 45 20 44 52 b8  cb
6d4c : 43 4b 45 4e 40 0d 18 0d  37
6d54 : 47 45 4e 45 52 49 45 52  a3
6d5c : 45 20 d7 4f 52 54 4c 49  1d
6d64 : 53 54 45 40 0d 0c 05 c1  04
6d6c : 4e 5a 41 48 4c 20 41 4c  a4
6d74 : 4c 45 52 20 d7 b6 52 54  21
6d7c : 45 52 20 3a 20 40 0d 0e  8e
6d84 : 05 d6 45 52 53 43 48 49  93
6d8c : 45 44 45 4e 45 20 d7 b6  31
6d94 : 52 54 45 52 20 3a 20 40  81
6d9c : 0d 18 03 d7 4f 52 54 20  8a
6da4 : 55 4e 42 45 4b 41 4e 4e  ee
6dac : 54 21 20 cc 45 52 4e 45  dd
6db4 : 4e 20 3f 20 28 4a 41 2f  1e
6dbc : 4e 45 49 4e 29 40 0d 18  c2
6dc4 : 08 d7 4f 52 54 20 4c b6  bb
6dcc : 53 43 48 45 4e 20 3f 20  9f
6dd4 : 28 4a 41 2f 4e 45 49 4e  28
6ddc : 29 40 0d 24 00 00 00 00  ed
6de4 : 00 00 00 00 00 00 00 00  e5
6dec : 00 00 00 00 00 00 00 00  ed
6df4 : 00 00 00 00 53 50 45 4c  5a
6dfc : 4c 2e 44 49 43 54 2e 41  ac
6e04 : 53 50 45 4c 4c 2e 44 49  34
6e0c : 43 54 2e 48 45 4c 50 52  ab
6e14 : 3a 53 50 45 4c 4c 2e 44  1d
6e1c : 49 43 54 2e 48 45 4c 50  62
6e24 : 3d 53 50 45 4c 4c 2e 44  30
6e2c : 49 43 54 2e 41 53 3a 53  30
6e34 : 50 45 4c 4c 2e 44 49 43  74
6e3c : 54 2e 48 45 4c 50 00 00  a9
6e44 : 00 00 00 00 00 00 00 00  45
6e4c : 00 00 00 00 00 00 00 00  4d
6e54 : 00 00 00 01 03 04 05 06  e5
6e5c : 07 08 09 0a 0b 0c 0d 0e  4c
6e64 : 0f 10 12 13 14 15 17 18  d9
6e6c : 1a 1b 1c 1d 1e 02 11 19  27
6e74 : 16 00 ff 00 ff 00 ff 00  8a
```

Listing 4. (Schluß)

```
1000 PRINT"{CLR,DOWN,RIGHT,CTRL-N}BITTE DI
     E DISKETTE FUER DAS                   <142>
1010 PRINT"{DOWN,RIGHT}WOERTERVERZEICHNIS
     EINLEGEN!                             <062>
1020 POKE 198,0:WAIT 198,1                 <230>
1030 FOR I=65 TO 90                        <135>
1040 A$=CHR$(I)                            <056>
1050 PRINT"{HOME,6DOWN,39SPACE,2UP}"       <024>
1060 PRINT"{RIGHT}VERZEICHNIS FUER BUCHSTA
     BE : ";A$                             <239>
1070 OPEN 1,8,1,"SPELL.DICT."+A$           <008>
1080 PRINT#1,CHR$(255);                    <036>
1090 GOSUB 10000                           <054>
1100 CLOSE 1                               <095>
1110 NEXT I                                <178>
1120 PRINT"{DOWN,RIGHT}VERZEICHNIS FERTIG  <136>
1130 END                                   <116>
10000 OPEN 15,8,15                         <252>
10010 INPUT#15,E,E$,T,S                    <077>
10020 PRINT"DISK-ERROR: ";E;E$;T;S         <208>
10030 CLOSE 15:RETURN                      <196>
```

Listing 5. »DICTIONARY.GEN« erzeugt die Wörterbuch-Dateien auf Diskette. Bitte mit dem Checksummer (Seite 159) eingeben.

Disk-Füller: Retter in der Not

Ab und zu passiert es, daß auf einer fast vollen Diskette nur ein paar Blöcke fehlen, um ein Programm zu speichern. Mit dem Disk-Füller können in Notfällen bis zu 17 zusätzliche Blöcke auf der Diskette genutzt werden.

Das Problem ist wohl jedem bekannt, der eine gut sortierte Programmsammlung hat: Alle thematisch gleichen Programme bis auf eines passen auf eine Diskettenseite. Und gerade dieses letzte Programm benötigt nur einige wenige Blöcke. Aber hier kann in Notfällen Abhilfe geschaffen werden. Auf der Spur 18, auf der sich unter anderem das Directory befindet, sind meist noch ein paar Blöcke frei, die das DOS der 1541 aber nicht zur Programmspeicherung zur Verfügung stellt. Mit dem Programm »Disk-Füller« (Listing 1) können diese freien Blöcke ausgenutzt werden. Zu diesem Zweck werden Blöcke von schon auf der Diskette befindlichen Programmen auf die freien Blöcke der Spur 18 umkopiert und die Blockzeiger angepaßt. Die ursprünglichen Programmblöcke stehen dann zur freien Verfügung.

Das Programm ist weitgehend selbsterklärend, hier aber einige wichtige Hinweise: Nach dem Laden darf das Programm erst mit RUN gestartet werden, wenn die zu behandelnde Diskette im Laufwerk liegt. Außerdem dürfen nur Disketten behandelt werden, deren BAM einwandfrei in Ordnung ist. Im Zweifelsfalle sollte vorher ein Validate mit OPEN 15,8,15,"V" durchgeführt werden.

Hier ist die genaue Funktionsweise des Programms:

Nach einer kurzen Erläuterung für den Benutzer werden in den Zeilen 1050 und 1060 (Listing) zwei Kanäle zum Laufwerk geöffnet. Gleichzeitig wird die eingelegte Diskette initialisiert. In Zeile 2000 werden drei Variablenfelder dimensioniert. Sie dienen zum Ablegen der Zeiger auf den jeweils nächsten Directoryblock sowie der Anzahl der Einträge in diesem Block. Mit Zeile 2010 beginnt dann eine Schleife, die alle Directory-Blöcke verfolgt, ihre Lage auf der Diskette in den dimensionierten Feldern speichert und ihre Anzahl in der Variablen B festhält. Ab Zeile 2500 wird die Anzahl der freien Einträge im letzten Directory-Block bestimmt. Dazu wird jeweils das Filetyp-Byte abgefragt. Hat es den Wert Null, ist dieser Eintrag noch frei. Sollte der Benutzer die Frage, ob der angezeigte Platz genügt, negativ beantworten, wird ein Block für weitere acht Einträge freigehalten. Als nächstes wird überprüft, welche Blöcke der Spur 18 noch frei sind. Dazu werden die entsprechenden Bytes der BAM in den Computerspeicher geholt und analysiert (Zeile 3120 bis 3230).

In Zeile 4000 beginnt das Verschieben schon belegter Blöcke in die Spur 18. Dazu werden in einer großen Schleife (bis Zeile 4680) alle Directory-Einträge abgearbeitet. Dies geschieht folgendermaßen: Der Zeiger auf den ersten Block des ersten Eintrags wird in den Variablen AS und AT zwischengespeichert, der zugehörige Block eingelesen und der Zeiger auf den Fortsetzungsblock ebenfalls gespeichert. Dann wird ein freier Block der Spur 18 gesucht und der alte Zeiger auf diesen umgelenkt. Der Computer gibt nun eine Protokollzeile aus. Dieser Block wird nun als belegt und der alte als frei gekennzeichnet. Als letztes wird endlich der Inhalt des zu verschiebenden Blocks auf den Block der Spur 18 geschrieben. Sollte nach dem ersten Eintrag immer noch Platz auf der Spur 18 sein, fährt das Programm mit dem zweiten Eintrag fort, bis die Spur 18 komplett belegt ist.

(U. Gerlach/sk)

```
100 REM   ***************************        <140>
110 REM   *                         *        <159>
120 REM   *                         *        <169>
130 REM   *     DISK   FUELLER      *        <115>
140 REM   *                         *        <189>
150 REM   *                         *        <199>
160 REM   *                         *        <209>
170 REM   *    VON  UWE  GERLACH    *        <039>
180 REM   *                         *        <229>
190 REM   *     BRUEHLSTR. 23       *        <186>
200 REM   *                         *        <249>
210 REM   *                         *        <003>
220 REM   *     6440  BEBRA 1       *        <164>
230 REM   *                         *        <023>
240 REM   *                         *        <033>
250 REM   *                         *        <043>
260 REM   *                         *        <055>
270 REM   *     IM APRIL 1985       *        <238>
280 REM   *                         *        <075>
290 REM   *                         *        <085>
300 REM   ***************************        <086>
400 :                                        <122>
500 :                                        <222>
1000 POKE 53280,0: POKE 53281,0: REM FARBE <068>
1010 PRINT CHR$(14);"{CLR,GREY 2,2SPACE}VC
      1541{4SPACE}D I S K{3SPACE}F U E L L
      E R":PRINT:PRINT:PRINT               <073>
1015 PRINT"DIESES PROGRAMM WURDE GESCHRIEB
     EN VON:":PRINT:PRINT                  <051>
1020 PRINT,"{LIG.BLUE,2SPACE}UWE GERLACH":
      PRINT,"{2SPACE}BRUEHLSTR. 23": PRINT
     : PRINT,"{2SPACE}6440 BEBRA 1{GREY 2}
     "                                     <080>
1025 PRINT:PRINT:PRINT"ES SCHAFFT AUF VOLL
     EN DISKETTEN NOCH"                    <012>
1030 PRINT"ETWAS PLATZ, INDEM ES DIE EIGEN
     TLICH"                                <149>
1035 PRINT"FUER DAS DIRECTORY VORGESEHENE
     SPUR 18"                              <209>
1040 PRINT"BELEGT.":PRINT:PRINT            <248>
1045 :                                     <005>
1050 OPEN 1,8,15,"I": REM  FUER BEFEHLE    <004>
1060 OPEN 2,8,2,"#" : REM  PUFFERKANAL     <091>
1090 :                                     <050>
1100 REM  ******************************   <096>
1110 REM    DIRECTORYLAENGE ERMITTELN      <193>
1120 REM  ******************************   <116>
1130 :                                     <090>
2000 DIM T(25): DIM S(25): DIM E(25)       <001>
2010 PRINT# 1, "U1:"2;0;18;0: REM  BAM     <085>
2020 GET# 2,T$,S$: REM  ZEIGER AUF DIR.    <111>
2030 T(0)=ASC(T$+CHR$(0)): REM  TRACK      <218>
2040 S(0)=ASC(S$+CHR$(0)): REM  SECTOR     <222>
2050 T=T(0): REM  TRACK  DIR. ANFANG       <167>
2060 S=S(0): REM  SECTOR DIR. ANFANG       <220>
2070 B=1: REM  BLOCKZAEHLER                <212>
2100 PRINT# 1, "U1:"2;0;T;S                <176>
2110 GET# 2,T$,S$: REM  ZEIGER             <065>
2120 T=ASC(T$+CHR$(0))                     <231>
2130 S=ASC(S$+CHR$(0))                     <172>
2140 IF T=0 OR T>35 OR S>20 OR B=25 THEN 2
     400:  REM  LETZTER DIRECTORY-BLOCK    <043>
2150 T(B)=T: S(B)=S: REM  ZEIGER MERKEN    <174>
2155 E(B-1)=8: REM  EINTRAEGE IM BLOCK     <111>
2160 B=B+1: REM  ZAEHLER ERHOEHEN          <082>
2190 GOTO 2100 : REM  NAECHSTER BLOCK      <174>
2400 M=B: REM  DIR. BLOCKZAHL              <228>
2500 F=0: REM  ZAEHLER FREIE EINTRAEGE     <071>
2510 FOR I=2 TO 226 STEP 32                <227>
2520 : PRINT# 1, "B-P:"2,I: REM ZEIGER     <006>
2530 : GET# 2,W$: REM  FILETYP             <176>
2540 : IF W$="" THEN F=F+1: REM  FREI      <091>
2590 NEXT I: REM  ALLE EINTR. IM BLOCK     <090>
2600 E(B-1)=8-F: REM  EINTRAEGE/BLOCK      <232>
2610 IF F=8 AND M=1 THEN 7800: REM LEER    <017>
2700 PRINT "GENUEGEN";F;"FREIE DIRECTORY-P
     LAETZE ? ";                           <031>
2750 GET A$: IF A$="" THEN 2750            <230>
```

```
2760 IF A$<>"N" THEN A$="J"                      <234>
2770 PRINT A$: PRINT: PRINT: PRINT              <213>
2790 :                                           <226>
3000 REM  *****************************          <218>
3010 REM    LEERE DIR. BLOECKE FESTST.           <168>
3020 REM  *****************************          <238>
3030 :                                           <212>
3050 PRINT# 1, "U1:"2;0;18;0: REM  BAM           <109>
3060 PRINT# 1, "B-P:"2,73: REM  SPUR 18          <008>
3100 B=0 : REM  BLOCKZAEHLER                     <210>
3110 L=19: REM  ZAHL LEERE DIR-BLOECKE           <082>
3120 DIM Z(25): REM  ZUSTAND DER BLOCKS          <004>
3130 FOR I=1 TO 3: REM  BYTES SPUR 18            <154>
3140 : GET# 2, W$: REM  BELEGUNGSCODE            <176>
3150 : W=ASC(W$+CHR$(0))                         <011>
3160 : FOR J=1 TO 8: REM  DUALSTELLEN            <207>
3170 :   W=W/2                                   <181>
3180 :   IF W=INT(W) THEN Z(B)=1                 <203>
3190 :   IF Z(B)=0 AND A$="N" THEN A$="": Z
     (B)=1: REM  EINEN BLOCK FREIHALTEN          <092>
3195 :   IF Z(B)=1 AND B<19 THEN L=L-1           <093>
3200 :   B=B+1                                   <071>
3085 :   W=INT(W)                                <134>
3220 : NEXT J                                    <076>
3230 NEXT I                                      <010>
3310 IF L<1 THEN 8000                            <076>
3390 PRINT: PRINT: PRINT "ES WERDEN NUN";L
     ;" BLOECKE FREIGESTELLT !": PRINT           <233>
3990 :                                           <156>
4000 REM  *****************************          <202>
4010 REM    FUELLEN DER DIR.- BLOECKE            <251>
4020 REM  *****************************          <222>
4021 :                                           <187>
4025 B=0: REM  BLOCKZAEHLER, SIEHE OBEN          <163>
4027 FOR E=0 TO E(B)-1: REM EINTR/BLOCK          <038>
4030 AT=T(B): REM  ALTER DIR-TRACK               <099>
4035 AS=S(B): NS=AS: REM  ALTER SECTOR           <166>
4040 IF AT=0 AND AS=0 THEN 7050                  <004>
4050 I=0                                         <173>
4060 Z=E*32+3: REM  STELLE DES ZEIGERS           <170>
4100 R=0: REM  MARKE DIR.-BLOCK AENDERN          <117>
4150 PRINT# 1, "U1:"2;0;AT;AS: REM  ALT          <111>
4160 PRINT# 1, "B-P:"2,Z: REM  ZEIGER            <151>
4170 GET# 2, T$,S$: REM  ZEIGER FORTS.           <185>
4180 T=ASC(T$+CHR$(0))                           <003>
4190 S=ASC(S$+CHR$(0))                           <200>
4191 IF T=18 THEN 4670: REM  BEARBEITET          <245>
4192 IF T=0 OR T>35 OR S>20 THEN Z=1: GOTO
      4520: REM  FILEENDE                        <211>
4200 FOR I=0 TO 18 STEP 10: REM  SUCHE NAC
     H FREIEM SECTOR IM SECTORABSTAND            <037>
4210 : IF Z(I)=0 THEN 4500: REM  LEER            <196>
4212 : IF I=9 THEN GOTO 4230                     <056>
4215 : IF I>8 THEN I=I-9: GOTO 4210              <096>
4220 NEXT I                                      <240>
4230 Z=-1: REM  MARKE: SPUR 18 VOLL              <231>
4300 GOTO 4520: REM  RUECK OHNE AENDER.          <214>
4500 PRINT# 1, "B-P:"2,Z: REM  ZEIGER            <237>
4510 PRINT# 2, CHR$(18);CHR$(I);: REM   ZE
     IGER VERBIEGEN                              <111>
4515 IF AT=18 AND AS=NS THEN R=1: REM   IN
      DEN BLOCK, AUS DEM GELESEN WURDE           <181>
4520 PRINT "{6SPACE}BLOCK";RIGHT$("{2SPACE
     }"+STR$(AT),3);",";RIGHT$("{2SPACE}"+S
     TR$(AS),3);                                 <252>
4525 IF R=1 THEN PRINT"{2SPACE}ZEIGER";E+1
     ;"GEAENDERT": GOTO 4570                     <023>
4530 PRINT "{2SPACE}-->{2SPACE}18,";RIGHT$
     ("{2SPACE}"+STR$(NS),3)                     <128>
4550 PRINT# 1, "B-A:"0,18,NS:REM IN BAM          <251>
4560 PRINT# 1, "B-F:"0,AT,AS: REM  ALTEN B
     LOCK FREIGEBEN                              <209>
4570 Z(I)=1: REM BLOCK IN LISTE BELEGEN          <254>
4580 PRINT# 1, "U2:"2;0;18;NS:REM RUEC           <235>
4650 AT=T: AS=S: NS=I: REM  NORM. BLOCK          <035>
4655 IF Z=-1 THEN GOTO 9000: REM  OK             <109>
4660 IF Z<>1 THEN Z=0: GOTO 4100                 <234>
4670 NEXT E: REM  NEUER DIR.-EINTRAG             <120>
4680 B=B+1: GOTO 4027: REM  WEITER MIT NAE
     CHSTEM DIRECTORYBLOCK                       <087>
6990 :                                           <108>
7000 REM  *****************************          <154>
7010 REM         FEHLERMELDUNGEN                 <130>
7020 REM  *****************************          <174>
7030 :                                           <148>
7050 PRINT: PRINT "ES WAREN NICHT GENUG BE
     LEGTE BLOCKS DA."                           <036>
7060 PRINT " SPUR 18 IST IMMERNOCH TEILWEI
     SE FREI!"                                   <099>
7090 GOTO 9000                                   <236>
7800 PRINT:PRINT:PRINT"DIE DISKETTE IST LE
     ER!"                                        <146>
7810 GOTO 9000                                   <194>
8000 PRINT "TUT MIR LEID, DA IST NICHTS ZU
      MACHEN."                                   <162>
8010 PRINT "DIE GESAMTE SPUR 18 IST VOLL B
     ELEGT !"                                    <160>
9000 PRINT                                       <212>
9010 CLOSE 1: CLOSE 2: END                       <099>
```

Listing 1. »Disk-Füller«. Bitte beachten Sie die Eingabehinweise auf Seite 159.

Schnelle RS232 ohne Hardware

Bisher gab es mit dem C64 schon Probleme bei einer Übertragungsrate von 1200 bit/s. Ein kleines Programm schafft Abhilfe. Mit 4800 bit/s kommt Power in die serielle Schnittstelle.

Im Handbuch des C64 steht gar nichts, im Programmierhandbuch nichts Genaues. Was ist mit der sogenannten »frei definierbaren« Übertragungsrate?

Die RS232 des C64 arbeitet mit einer bitseriellen, asynchronen Übertragung. Das heißt ein Byte wird in Bits zerteilt und diese dann über eine Leitung in Form von Spannungswechseln gesendet. Eine asynchrone Datenübermittlung verwendet Kennzeichen für Byteanfang und Byteende. Bei manchen Computern können die Daten wunschweise auch synchron übermittelt werden. Der Computer sendet dann keine Start- oder Stoppbit mehr, es werden nur noch die Daten übertragen. Vorteil: Die Übertragungszeit verkürzt sich, da die Steuerbits entfallen.

Beim C 64 ist die Übertragung rein softwaremäßig gesteuert. Er setzt dazu den CIA #2 ein. Dieser Baustein besitzt zwei 16-Bit-Intervall-Timer, die von einem bestimmten Wert auf Null zählen und dann einen NMI auslösen können. Das nutzt die RS232-Schnittstelle aus. Die Software holt den Wert für den Timer B, der als Interruptgenerator dient, aus einer im ROM gespeicherten Tabelle. Es existieren zwei verschiedene Tabellen, aber dazu später mehr. Diese Werte werden in den Timer und in zwei Speicherstellen gebracht. Danach startet der Timer.

RS232 mit 4800 bit/s

Da beim C 64 das Senden und Empfangen über RS232 interruptgesteuert geschieht, werden zwei Puffer (Zwischenspeicher) benötigt. Einer der beiden Puffer steht für die zu sendenden Daten bereit. Die Daten werden nur in den Puffer geschrieben. Das Senden geschieht mit Hilfe des NMI-Interrupts, weil man ihn (im Gegensatz zum IRQ) nicht sperren kann. So werden die Daten in jedem Fall gesendet, es sei denn, man stoppt den Timer oder verbiegt den NMI-Vektor. Da die Daten in den Puffer geschrieben werden und das Beschreiben des Puffers normalerweise schneller geht als das Senden der Zeichen, sollte man, nachdem ein Zeichen gesendet wurde, das nächste Zeichen vorbereiten und warten, bis der Puffer frei ist. Beachtet man dies nicht, könnte versehentlich ein noch nicht gesendetes Zeichen überschrieben werden. Die Folge wäre eine fehlerhafte

Übertragung. Das Empfangen geschieht auf ähnliche Weise, denn die RDX-Leitung (hier gelangen die Daten vom anderen Computer zum C 64) ist nicht nur mit dem Port B des CIA #2 verbunden, sondern auch mit dem Eingang FLAG der CIA, der einen NMI auslösen kann. Das Empfangen geschieht also unabhängig vom Programm, daher werden die Daten in den Empfangspuffer geschrieben, der genauso groß ist wie der Sendepuffer: 256 Byte. Man sollte sich also darum kümmern, daß die Zeichen rechtzeitig aus dem Puffer gelesen werden. Durch die Kernelroutine READST (ab $FFB7), oder beim C 64 auch durch Auslesen der Speicherstelle $90, läßt sich der Status abfragen. Bei gesetztem Bit 2 lief der Empfangspuffer über. Um nun andere Übertragungsraten zu benutzen, könnten natürlich die Werte in den Timer und die entsprechenden Werte in die beiden Speicherstellen geschrieben werden. Dies hätte nur den großen Nachteil, daß die so erstellten Programme schwer an andere (Commodore-)Computer anzupassen wären, da diese nicht die gleichen Speicherstellen benutzen. Aber es geht auch viel eleganter!

Die RS232-Schnittstelle öffneten wir bisher folgendermaßen:

```
lda #2       ;Logische Filenummer = 2
ldx #2       ;Gerätenummer der RS232
ldy #3       ;Sekundäradresse (ohne Funktion)
jsr setpar   ;Werte setzen
lda #2       ;Länge des Filenamens
ldx #<fadr   ;Adresse des Filenamens (low)
ldy #>fadr   ;(high)
jsr setnam   ;Name für OPEN setzen
jsr open     ;Datei öffnen
ldx #<ebuf   ;Zeiger für Empfangspuffer
ldy #>ebuf   ;und
stx $f7
sty $f8
ldx #<abuf   ;Ausgabepuffer setzen
ldy #>abuf
stx $f9
sty $fa
fadr .byte %00000110 ;Wert für Steuerregister
     .byte %00000000 ;Wert für Befehlsregister
```

Übrigens werden die Zeiger auf die Puffer auch vom Betriebssystem gesetzt. Dieses legt die Puffer an das obere Basic-RAM-Ende. Dies ist auch der Grund, weswegen beim Öffnen der Schnittstelle ein CLR ausgeführt wird, denn im gleichen Speicherbereich liegen ja auch die Stringvariablen. Darum legen wir die Puffer in uns angenehme Speicherbereiche, was man entweder vor oder nach dem OPEN-Befehl machen kann. Vorher deswegen, weil das Betriebssystem die Pufferadressen erst festlegt, nachdem es in den Speicherstellen nachgesehen hat, ob das vom Programm schon gemacht wurde. Ist das High-Byte der Pufferzeiger ungleich Null, wird davon ausgegangen, daß die Zeiger bereits auf die Puffer deuten. Deshalb wird nichts mehr verändert.

Und nun geht's los!

Der Filename kann auch vier Zeichen lang sein. Die ersten beiden Zeichen behalten ihre Bedeutung bei, aber die nächsten beiden beinhalten den Wert für die Übertragungsrate.

Der Filename sieht also wie folgt aus:
»Steuerregister« »Befehlsregister« »Übertragungsrate-low« »Übertragungsrate-high«

Wie gesagt, das untere Nibble vom »Steuerregister« muß %0000 sein! Die Werte für »Übertragungsrate-low« und »Übertragungsrate-high« berechnet man wie folgt:

```
Übertragungsrate-high = INT(Prozessortakt/
Übertragungsrate/2-100)/256)
Übertragungsrate-low= INT(Prozessortakt/
Übertragungsrate/2-100-Übertragungsrate-highx256)
```

Der Prozessortakt beträgt bei der deutschen PAL-Version 985259 Hertz, bei der amerikanischen NTSC-Version 1022730 Hertz. Das heißt nichts anderes, als daß die Programme auf amerikanischen Geräten schneller laufen als auf deutschen. Da dies natürlich auch Auswirkungen auf die Übertragungskapazität hat, sind im Betriebssystem zwei Tabellen für den Timer B vorhanden. Wenn Sie nun Programme mit selbstdefinierten Übertragungsraten schreiben, sollten Sie immer daran denken, daß diese Programme entweder nur auf dem deutschen oder dem amerikanischen C 64 richtig laufen, wenn Sie nur eine Tabelle verwenden. Durch Auslesen der Speicherstelle $02A6 können Sie ganz einfach feststellen, ob Sie eine PAL- oder eine NTSC-Version vor sich haben. Bei einer PAL-Version steht in besagter Speicherstelle der Wert 1, bei NTSC eine Null.

Steuerregister	Befehlsregister
Bit 7:	Bit 7,6,5:
0 1 Stoppbit	000 keine Parität
1 2 Stoppbits	001 gerade Parität
	011 ungerade Parität
	101 8. Datenbit = 1
	111 8. Datenbit = 0
Bit 5,6:	Bit 4:
00 8 Datenbits	0 Vollduplex
01 7 Datenbits	1 Halbduplex
10 6 Datenbits	
11 5 Datenbits	
Bit 0,1,2,3:	Bit 0 (Handshake):
0000 selbstdef. Übertr.	0 3-Line (Software-)
0001 50 bit/s	1 X-Line (Hardware-)
0010 75 bit/s	
0011 110 bit/s	
0100 134,5 bit/s	
0101 150 bit/s	
0110 300 bit/s	
0111 600 bit/s	
1000 1200 bit/s	
1001 1800 bit/s	
1010 2400 bit/s	

Tabelle 1. Übersicht der Werte für Steuer- und Befehlsregister

Beispiel:

Die Übertragungsrate soll 1800 bit/s betragen. Also erhalten wir für »Übertragungsrate-high«: INT((985250/

```
Name : seq-trans.ob        cdØØ cded
----------------------------------------
cdØØ : 2Ø fd ae 2Ø 9e b7 86 Ø2   94
cdØ8 : 2Ø fd ae 2Ø 9e ad 2Ø a6   fc
cd1Ø : b6 c9 Ø5 bØ Ø3 4c Ø8 af   14
cd18 : a6 22 a4 23 2Ø bd ff a9   aØ
cd2Ø : Ø1 a2 Ø8 aØ Ø2 2Ø ba ff   95
cd28 : 2Ø cØ ff 2Ø b7 ff fØ Ø8   fb
cd3Ø : 2Ø cc ff a9 Ø1 4c c3 ff   6d
cd38 : a2 Ø1 2Ø c6 ff 2Ø b7 ff   1c
cd4Ø : dØ ee a2 Ø1 aØ Ø8 86 22   f9
cd48 : 84 23 2Ø cf ff aØ ØØ 91   88
cd5Ø : 22 e6 22 dØ Ø2 e6 23 2Ø   ac
cd58 : b7 ff fØ ee 29 4Ø fØ dØ   23
cd6Ø : 2Ø 3Ø cd a9 Ø1 a2 Ø2 aØ   af
cd68 : Ø3 2Ø ba ff a9 Ø4 a2 e5   3b
cd7Ø : aØ cd 2Ø bd ff 2Ø cØ ff   bb
cd78 : a2 Ø3 bd e9 cd 95 f7 ca   47
cd8Ø : 1Ø f8 a2 Ø1 2Ø c9 ff a2   6b
cd88 : Ø1 aØ Ø8 86 14 84 15 ad   c1
cd9Ø : Øe dc 29 fe 8d Øe dc aØ   34
cd98 : ØØ b1 14 24 Ø2 3Ø 15 c9   84
cdaØ : 41 9Ø Øf c9 8Ø bØ Ø4 Ø9   d6
cda8 : 2Ø dØ Ø7 c9 cØ 9Ø Ø3 38   38
cdbØ : e9 8Ø 29 7f aa ad a1 Ø2   b6
cdb8 : 29 Ø1 dØ f9 8a 2Ø d2 ff   ca
cdcØ : e6 14 dØ Ø2 e6 15 a5 22   17
cdc8 : c5 14 dØ cb a5 23 c5 15   fa
cddØ : dØ c5 ad a1 Ø2 29 Ø1 dØ   31
cdd8 : f9 2Ø 3Ø cd ad Øe dc Ø9   78
cdeØ : Ø1 8d Øe dc 6Ø ØØ ØØ Ø2   d1
cde8 : ØØ ØØ ce ØØ cf ff ØØ ff   99
```

Listing 1. RS232 mit 4800 bit/s. Bitte mit dem MSE (Seite 159) eingeben.

4800/2-100)/256)=0 und für »Übertragungsrate-low«: INT(985250/4800/2-100-0x256)=2. Der Filename lautet also: %XXXX0000, %XXXXXXXX, %00000010, %00000000 Achtung: Für die X-Bits setzen Sie die entsprechenden Werte aus der Tabelle 1 ein.

Übungen:

Wir möchten ein sequentielles File mit 4800 bit/s, 7 Datenbits, 1 Stoppbit, 3-Line-Handshake und keiner Parität senden. Also: Zuerst das sequentielle File in den Speicher laden. Jetzt müssen wir noch die RS232-Schnittstelle öffnen, wobei wir auf die Abfrage auf PAL- oder NTSC-Version verzichten. Das Programm zum Öffnen ist bis auf den Filenamen und dessen Länge von dem altbekannten nicht zu unterscheiden.

```
lda #2      ;logische Filenummer = 2
ldx #2      ;Gerätenummer der RS232
ldy #3      ;Sekundäradresse
jsr setlfs  ;Werte setzen
lda #4      ;der Filename ist 2 Zeichen lang
ldx #<fadr  ;Adresse des Filenamens (low)
ldy #>fadr  ;(high)
jsr setnam  ;Name für OPEN setzen
jsr open    Datei öffnen
ldx #2      ;Datei mit der logischen Filenummer
jsr chkout  ;#2 als Standardausgabegerät
```

Nun müssen Sie einen Zeiger auf den Anfang des in den Speicher geladenen sequentiellen Files setzen und die Pufferzeiger entsprechend ändern.

```
(Zeiger »pt« auf Dateianfang setzen)
sende ldy #0     ;aktuelles Zeichen aus dem
      lda (pt),y ;File in den Akku holen
(Zeichen nun bearbeiten, zum Beispiel in ASCII-Code
wandeln)
      tax        ;Zeichen merken
wloop lda $02a1  ;ist das vorherige Zeichen
      and #1     ;bereits gesendet worden?
      bne wloop  ;nein: =>warten
      txa        ;gemerktes Zeichen zurückholen
      jsr chrout ;auf RS232 ausgeben
```

Der Zeiger »PT« muß nun erhöht und auf Fileende getestet werden. Ist die Datei noch nicht zu Ende, muß wieder zum Label SENDE gesprungen werden.

```
      jsr clrchn ;Standardwerte Ein-/Ausgabe
      lda #2     ;setzen und Datei schließen
      jsr close
      rts        ;das wär's!
fadr  .byte %00100000
      .byte %00000000
      .byte %00000010
      .byte %00000000
```

Das folgende Programm (Listing 1, bitte mit dem MSE eingeben) sendet sequentielle Files mit 4800 bit/s, 8 Datenbits, 1 Stoppbit, keine Parität, Vollduplex und 3-Line Handshake.

Die Syntax lautet: SYS 52480,konv-flag," Name"

Wenn konv-flag = 0, werden die Zeichen erst in ASCII-Code gewandelt und dann gesendet. Bei konv-flag = 128 erfolgt die Übertragung unverändert.

Mit dem Namen ist der Titel der sequentiellen Datei gemeint. »,S,R"« wird vom Programm nicht automatisch ergänzt. Sie müssen es also noch an den Namen anhängen.

Beispiel:

SYS 52480,128," TEST,S,R"

Die Datei TEST wird unkonvertiert gesendet.

(Martin Müller/N. Heusler/kn)

Assembler-Listings übersichtlich ausgedruckt

Das Drucken einspaltiger Assembler- oder Disassembler-Listings ist reine Papiervergeudung. Die Übersichtlichkeit des Ausdrucks leidet ebenfalls. Der »Multi-Lister« bietet für den Star NL-10 und Epson-kompatible Drucker eine wirkungsvolle Lösung.

Wenn man ein Assembler-Listing mit dem Hypra-Ass oder dem SMON ausdruckt, erhält man einen Berg von Papier und ärgert sich, daß das Papier nur zur Hälfte bedruckt ist. Der »Multi-Lister« (Listing 1) schafft noch mehr: Er druckt bis zu sechs Spalten pro Seite.

Um den Multi-Lister zu verwenden, muß der zu druckende Text beziehungsweise das Assembler-Listing zuerst als sequentielles File auf Diskette erzeugt werden. Dies klingt komplizierter als es ist. Um beispielsweise ein Disassembler-Listing des SMON als sequentielles File auf Diskette zu erzeugen, geht man folgendermaßen vor: SMON laden.

```
NEW
OPEN 1,8,2,''SMON.TXT,S,W''
CMD 1
SYS 49152
D C000 D000
X
PRINT#1 : CLOSE1
```

Zuerst wird eine sequentielle Datei geöffnet. Anschließend wird die Ausgabe mit CMD 1 auf diese Datei umgelenkt. Nun startet man den SMON und gibt mit »D C000 D000« den Bereich von $C000 bis $D000 disassembliert aus. Nach dem CMD-Befehl sind Ihre Eingaben nicht mehr in gewohnter Weise auf dem Bildschirm sichtbar. Davon sollten Sie sich jedoch nicht stören lassen. Wenn der Computer sich mit READY zurückmeldet, verläßt man den SMON mit ⟨X⟩ und schließt die Datei.

Nun lädt man den Multi-Lister und startet ihn mit RUN.

Das Programm meldet sich mit dem Hauptmenü. Im Anschluß daran wählt man durch Drücken der entsprechenden Taste einen der fünf Menüpunkte aus. Mit ⟨1⟩ wird der Name der Textdatei abgefragt, die ausgedruckt werden soll. Zur Überprüfung des Textes stellt man mit ⟨2⟩ die Ausgabe auf den Bildschirm um. Hierbei wird aber nicht das beabsichtigte Druckformat eingehalten, da dieser Punkt nur zur Kontrolle dient.

Mit der Taste ⟨3⟩ gelangt man in ein Untermenü, in dem man das Format des Ausdrucks verändern kann. Sie kön-

nen dann zwischen der Schriftart »Normal« und »Micro« wählen, die Zeichen pro Spalte, und die Zahl der Spalten festlegen. Die Zahl der möglichen Spalten errechnet das Programm aus der von Ihnen angegebenen Spaltenbreite. Die entsprechenden Einstellungen werden im Hauptmenü angezeigt.

Ist alles richtig eingestellt, öffnen Sie mit ⟨4⟩ die Listing-Datei zum Lesen. Eventuelle Disk-Fehler werden dabei abgefangen. Anschließend druckt das Programm das Listing aus (Bild 1). Dabei kann man mit ⟨←⟩ den Druckvorgang abbrechen.

Mit ⟨5⟩ das Programm verlassen. Ein Neustart ist mit RUN problemlos möglich.

Das Programm ist speziell für den NL-10 geschrieben und nutzt die Fähigkeit des Schubtraktors, das Papier auch wieder zurückzuziehen. Das Programm funktioniert also prinzipiell auch auf Epson-kompatiblen Druckern, die solche Fähigkeiten haben.

Hier die wichtigsten Zeilen für die Druckeransteuerung:
1740 Reset und Condensed-Schrift
1750 Blatt Länge aus 12 Inch
1770 Line-Feed auf 6/72 Inch
1780 Superscript
1820 Linken Rand einstellen
1890 Reverse Form-Feed
1900 Form-Feed

Das kleine Maschinencode-Programm, das ab Zeile 2290 in Form von DATA-Zeilen vorliegt, ist eine Inform-Routine, die den schwachen INPUT #-Befehl im Basic des C 64 ersetzt. Diese Routine ist frei verschiebbar (Startadresse $AD).

Um das Programm etwa für den Epson FX-85 anzupassen, muß Zeile 1890 folgendermaßen geändert werden:

```
1890 FOR RU = 1 TO 10: PRINT 2#, CHR$ (27) CHR$
(106) CHR$(216);: NEXT RU: GOTO 1820
```

Der FX-85 kennt nämlich nicht den reversen Form-Feed, sondern nur einen reversen Zeilenvorschub, der in n/216-Zoll-Schritten eingestellt werden kann. Durch die FOR-NEXT-Schleife wird genau dies 10mal veranlaßt. Um der besseren Lesbarkeit willen kann man beim FX-85 durch Einfügen der Zeile:

```
1745 PRINT#2,CHR$(27) CHR$(71)
```

zusätzlich Doppeldruck einschalten.

(Andreas Elvers/U.Beenen)

```
READY.

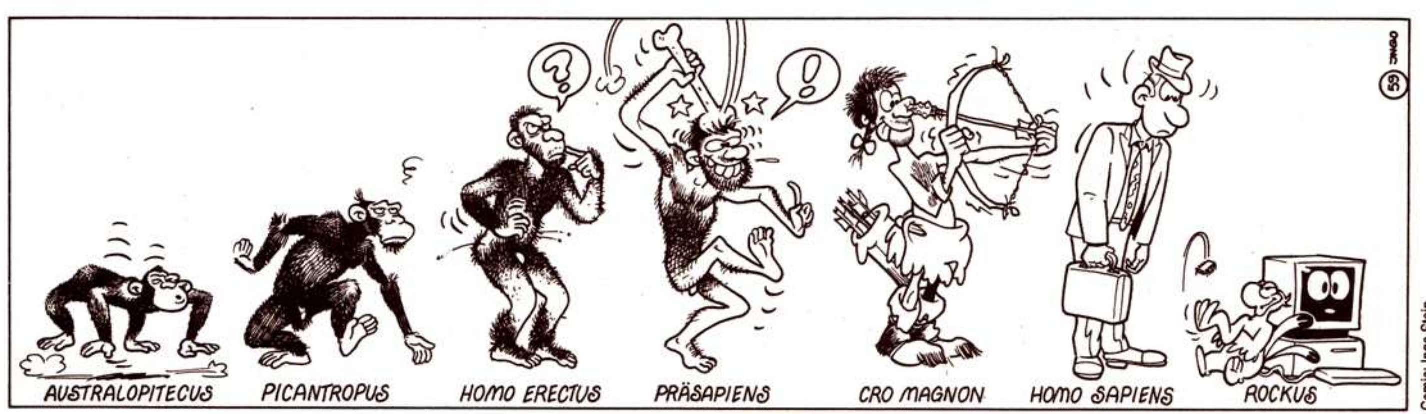
  PC  SR AC XR YR SP  NV-BDIZC
;C00B B0 C2 00 00 F2  10110000
.
:0801 1C 08 E8 03 8F 20 2A 2A
:0809 2A 2A 2A 2A 2A 2A 2A 2A
:0811 2A 2A 2A 2A 2A 2A 2A 2A
:0819 2A 2A 00 37 08 F2 03 8F
:0821 20 2A 20 20 20 4D 55 4C
:0829 54 49 2D 4C 49 53 54 45
:0831 52 20 20 20 2A 00 52 08
:0839 FC 03 8F 20 2A 20 20 20
:0841 28 43 29 20 20 31 39 38
:0849 37 20 42 59 20 20 20 2A
:0851 00 6D 08 06 04 8F 20 2A
:0859 20 20 41 4E 44 52 45 41
:0861 53 20 45 4C 56 45 52 53
:0869 20 20 2A 00 88 08 10 04
:0871 8F 20 2A 2A 2A 2A 2A 2A
:0879 2A 2A 2A 2A 2A 2A 2A 2A
:0881 2A 2A 2A 2A 2A 2A 00 8E
:0889 08 1A 04 3A 00 9B 08 24
:0891 04 41 44 B2 33 36 38 36
:0899 34 00 B9 08 2E 04 81 20
:08A1 49 B2 41 44 20 A4 20 41
:08A9 44 AA 37 37 3A 87 20 41
:08B1 3A 97 49 2C 41 3A 82 00
:08B9 D9 08 38 04 41 55 24 B2
:08C1 22 C4 52 55 43 4B 45 52
:08C9 22 3A 54 58 24 B2 22 12
:08D1 4B 45 49 4E 45 92 22 00
:08D9 EF 08 42 04 46 4F 24 B2
:08E1 22 4E 4F 52 4D 41 4C 2F
:08E9 34 30 2F 33 22 00 FE 08
:08F1 4C 04 5A 53 B2 34 30 3A
:08F9 53 50 B2 33 00 0C 09 56
:0901 04 9F 32 35 35 2C 38 2C
:0909 31 35 00 2E 09 60 04 97
:0911 35 33 32 38 30 2C 30 3A
:0919 97 35 33 32 38 31 2C 31
:0921 31 3A 99 C7 28 31 34 29
:0929 C7 28 38 29 00 34 09 6A
:0931 04 3A 00 69 09 74 04 99
:0939 22 93 9B 12 20 20 20 20
:0941 20 20 20 20 20 20 20 20
:0949 CD 55 4C 54 49 2D CC 49
:0951 53 54 45 52 20 D6 31 2E
:0959 30 20 20 20 20 20 20 20
:0961 20 20 20 20 92 22 3B 00
:0969 9A 09 7E 04 99 22 05 A3
:0971 A3 A3 A3 A3 A3 A3 A3 A3
:0979 A3 A3 A3 A3 A3 A3 A3 A3
:0981 A3 A3 A3 A3 A3 A3 A3 A3
:0989 A3 A3 A3 A3 A3 A3 A3 A3
:0991 A3 A3 A3 A3 A3 A3 A3 22
:0999 00 BD 09 88 04 99 A6 31
:09A1 30 29 22 11 11 11 D3 49
:09A9 45 20 48 41 42 45 4E 20
:09B1 44 49 45 20 D7 41 48 4C
:09B9 20 21 22 00 DE 09 92 04
:09C1 99 A6 31 30 29 22 90 A3
:09C9 A3 A3 A3 A3 A3 A3 A3 A3
:09D1 A3 A3 A3 A3 A3 A3 A3 A3
:09D9 A3 A3 A3 22 00 FF 09 9C
:09E1 04 99 22 11 9E 20 20 31
:09E9 2E 29 20 D4 45 58 54 44
:09F1 41 54 45 49 20 20 20 3A
:09F9 22 3B 54 58 24 00 1E 0A
:0A01 A6 04 99 22 20 20 32 2E
:0A09 29 20 C1 55 53 47 41 42
:0A11 45 20 41 55 46 20 3A 22
:0A19 3B 41 55 24 00 3D 0A B0
:0A21 04 99 22 20 20 33 2E 29
:0A29 20 C6 4F 52 4D 41 54 49
:0A31 45 52 55 4E 47 3A 22 3B
:0A39 46 4F 24 00 55 0A BA 04
:0A41 99 22 20 20 34 2E 29 20
:0A49 C4 52 55 43 4B 45 4E 20
:0A51 21 21 22 00 67 0A C4 04
:0A59 99 22 20 20 35 2E 29 20
:0A61 C5 4E 44 45 22 00 9B 0A
:0A69 CE 04 99 22 90 A4 A4 A4
:0A71 A4 A4 A4 A4 A4 A4 A4 A4
:0A79 A4 A4 A4 A4 A4 A4 A4 A4
:0A81 A4 A4 A4 A4 A4 A4 A4 A4
:0A89 A4 A4 A4 A4 A4 A4 A4 A4
:0A91 A4 A4 A4 A4 A4 22 00 C5
:0A99 0A D8 04 99 22 05 11 20
:0AA1 20 20 20 28 43 29 20 CA
:0AA9 55 4C 59 20 31 39 38 37
:0AB1 20 42 59 20 C1 4E 44 52
:0AB9 45 41 53 20 C5 4C 56 45
:0AC1 52 53 22 00 E8 0A E2 04
:0AC9 99 22 90 20 20 20 20 20
:0AD1 20 20 20 20 20 20 CE 45
:0AD9 44 44 45 52 53 54 45 20
:0AE1 D7 45 47 20 36 22 00 09
:0AE9 0B EC 04 99 22 20 20 20
:0AF1 20 20 20 20 20 20 20 20
:0AF9 20 33 31 33 39 20 20 CB
:0B01 41 54 45 4D 49 4E 22 00
:0B09 39 0B F6 04 99 22 AF AF
:0B11 AF AF AF AF AF AF AF AF
:0B19 AF AF AF AF AF AF AF AF
:0B21 AF AF AF AF AF AF AF AF
:0B29 AF AF AF AF AF AF AF AF
:0B31 AF AF AF AF AF AF 22 00
:0B39 3F 0B 00 05 3A 00 5B 0B
:0B41 0A 05 A1 41 24 3A 8B 41
:0B49 24 B3 22 31 22 B0 41 24
:0B51 B1 22 35 22 A7 31 32 39
:0B59 30 00 69 0B 14 05 97 32
:0B61 31 34 2C 31 35 3A 99 00
:0B69 A3 0B 1E 05 81 49 B2 31
:0B71 A4 35 3A 99 22 20 20 20
:0B79 20 20 20 20 20 20 20 20
:0B81 20 20 20 20 20 20 20 20
:0B89 20 20 20 20 20 20 20 20
:0B91 20 20 20 20 20 20 20 20
:0B99 20 20 20 20 20 22 3B 3A
:0BA1 82 00 B1 0B 28 05 97 32
:0BA9 31 34 2C 31 35 3A 99 00
:0BB1 D5 0B 32 05 91 C5 28 41
:0BB9 24 29 8D 31 33 38 30 2C
:0BC1 31 34 34 30 2C 31 34 39
:0BC9 30 2C 31 36 37 30 2C 32
:0BD1 30 39 30 00 DF 0B 3C 05
:0BD9 89 31 31 34 30 00 E5 0B
:0BE1 46 05 3A 00 EB 0B 50 05
:0BE9 3A 00 F1 0B 5A 05 3A 00
:0BF1 10 0C 64 05 99 22 11 11
:0BF9 90 CE 41 4D 45 20 44 45
:0C01 52 20 D4 45 58 54 44 41
:0C09 54 45 49 3A 22 3B 00 20
:0C11 0C 6E 05 4C 41 B2 31 36
:0C19 3A 8D 32 31 34 30 00 2D
:0C21 0C 78 05 54 58 24 B2 54
:0C29 24 3A 8E 00 33 0C 82 05
:0C31 3A 00 39 0C 8C 05 3A 00
:0C39 3F 0C 96 05 3A 00 65 0C
:0C41 A0 05 8B 41 55 24 B2 22
:0C49 C4 52 55 43 4B 45 52 22
:0C51 A7 41 55 24 B2 22 C2 49
:0C59 4C 44 53 43 48 49 52 4D
:0C61 22 3A 8E 00 79 0C AA 05
:0C69 41 55 24 B2 22 C4 52 55
:0C71 43 4B 45 52 22 3A 8E 00
:0C79 7F 0C B4 05 3A 00 85 0C
:0C81 BE 05 3A 00 8B 0C C8 05
:0C89 3A 00 B5 0C D2 05 99 22
:0C91 11 D3 43 48 52 49 46 54
:0C99 41 52 54 20 28 CE 29 4F
:0CA1 52 4D 41 4C 2F 28 CD 29
:0CA9 49 43 52 4F 3A 20 20 9D
:0CB1 9D 22 3B 00 C4 0C DC 05
:0CB9 4C 41 B2 31 3A 8D 32 31
:0CC1 34 30 00 E4 0C E6 05 8B
:0CC9 54 24 B2 22 4E 22 A7 46
:0CD1 4F 24 B2 22 4E 4F 52 4D
:0CD9 41 4C 2F 22 3A 89 31 35
:0CE1 34 30 00 04 0D F0 05 8B
:0CE9 54 24 B2 22 4D 22 A7 46
:0CF1 4F 24 B2 22 20 4D 49 43
:0CF9 52 4F 2F 22 3A 89 31 35
:0D01 34 30 00 15 0D FA 05 99
:0D09 22 91 91 22 3B 3A 89 31
:0D11 34 39 30 00 41 0D 04 06
:0D19 99 22 11 DA 45 49 43 48
:0D21 45 4E 20 50 52 4F 20 D3
:0D29 50 41 4C 54 45 20 28 31
:0D31 2D 31 33 34 29 20 3A 20
:0D39 20 20 9D 9D 9D 22 3B 00
:0D41 50 0D 0E 06 4C 41 B2 33
:0D49 3A 8D 32 31 34 30 00 7A
:0D51 0D 18 06 5A 53 B2 B5 28
:0D59 C5 28 54 24 29 29 3A 8B
:0D61 5A 53 B3 31 B0 5A 53 B1
:0D69 31 33 34 A7 99 22 91 91
:0D71 22 3B 3A 89 31 35 34 30
```

Bild 1. Ein Ausdruck mit dem »Multi-Lister«. Er druckt bis zu sechs Spalten auf einer Seite.

```
1000 REM ********************          <232>
1010 REM *    MULTI-LISTER    *        <195>
1020 REM *    (C)  1987 BY    *        <206>
1030 REM *  ANDREAS ELVERS    *        <161>
1040 REM ********************          <018>
1050 :                                 <010>
1060 AD=36864                          <043>
1070 FOR I=AD TO AD+77:READ A:POKE I,A:NEX
     T                                 <141>
1080 AU$="DRUCKER":TX$="{RVSON}KEINE{RVOFF
     }"                                <200>
1090 FO$="NORMAL/40/3"                 <065>
1100 ZS=40:SP=3                        <135>
1110 OPEN 255,8,15                     <239>
1120 POKE 53280,0:POKE 53281,11:PRINT CHR$
     (14)CHR$(8)                       <158>
1130 :                                 <090>
1140 PRINT"{CLR,GREY 3,RVSON,12SPACE}MULTI
     -LISTER V1.0{11SPACE,RVOFF}";     <153>
1150 PRINT"{WHITE}TTTTTTTTTTTTTTTTTTTTTTTTTT
     TTTTTTTTTTTTTTTTTT"               <125>
1160 PRINT SPC(10)"{3DOWN}SIE HABEN DIE WA
     HL !"                             <015>
1170 PRINT SPC(10)"{BLACK}TTTTTTTTTTTTTTTTTT
     TTTT"                             <161>
1180 PRINT"{DOWN,YELLOW,2SPACE}1.) TEXTDAT
     EI{3SPACE}:";TX$                  <135>
1190 PRINT"{2SPACE}2.) AUSGABE AUF :";AU$  <001>
1200 PRINT"{2SPACE}3.) FORMATIERUNG:";FO$  <178>
1210 PRINT"{2SPACE}4.) DRUCKEN !!"     <103>
1220 PRINT"{2SPACE}5.) ENDE"           <131>
1230 PRINT"{BLACK}@@@@@@@@@@@@@@@@@@@@@@@@@@
     @@@@@@@@@@@@@@@@@"                <139>
1240 PRINT"{WHITE,DOWN,4SPACE}(C) JULY 198
     7 BY ANDREAS ELVERS"              <170>
1250 PRINT"{BLACK,11SPACE}NEDDERSTE WEG 6" <038>
1260 PRINT"{12SPACE}3139{2SPACE}KATEMIN"   <149>
1270 PRINT"PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
     PPPPPPPPPP"                       <121>
1280 :                                 <242>
1290 GET A$:IF A$<"1"OR A$>"5"THEN 1290    <183>
1300 POKE 214,15:PRINT                 <088>
1310 FOR I=1 TO 5:PRINT"{40SPACE}";:NEXT   <044>
1320 POKE 214,15:PRINT                 <108>
1330 ON VAL(A$)GOSUB 1380,1440,1490,1670,2
     090                               <148>
1340 GOTO 1140                         <154>
1350 :                                 <056>
1360 :                                 <066>
1370 :                                 <076>
1380 PRINT"{2DOWN,BLACK}NAME DER TEXTDATEI
     :";                               <057>
1390 LA=16:GOSUB 2140                  <007>
1400 TX$=T$:RETURN                     <171>
```

```
1410 :                                        <116>
1420 :                                        <126>
1430 :                                        <136>
1440 IF AU$="DRUCKER"THEN AU$="BILDSCHIRM"
     :RETURN                                  <164>
1450 AU$="DRUCKER":RETURN                     <025>
1460 :                                        <166>
1470 :                                        <176>
1480 :                                        <186>
1490 PRINT"{DOWN}SCHRIFTART (N)ORMAL/(M)IC
     RO:{2SPACE,2LEFT}";                      <229>
1500 LA=1:GOSUB 2140                          <171>
1510 IF T$="N"THEN FO$="NORMAL/":GOTO 1540 <053>
1520 IF T$="M"THEN FO$=" MICRO/":GOTO 1540 <096>
1530 PRINT"{2UP}";:GOTO 1490                  <054>
1540 PRINT"{DOWN}ZEICHEN PRO SPALTE (1-134
     ) :{3SPACE,3LEFT}";                      <000>
1550 LA=3:GOSUB 2140                          <031>
1560 ZS=INT(VAL(T$)):IF ZS<1 OR ZS>134 THE
     N PRINT"{2UP}";:GOTO 1540                <172>
1570 FO$=FO$+MID$(STR$(ZS),2,3)+"/"           <054>
1580 MS=INT(136/(ZS+2))                       <109>
1590 PRINT"{DOWN}WIEVIELE SPALTEN (1-";MS;
     "{LEFT}) :{SPACE,LEFT}";                 <028>
1600 LA=2:GOSUB 2140                          <049>
1610 SP=VAL(T$)                               <178>
1620 IF SP<1 OR SP>MS THEN PRINT"{2UP}";:G
     OTO 1590                                 <114>
1630 FO$=FO$+T$:RETURN                        <044>
1640 :                                        <092>
1650 :                                        <102>
1660 :                                        <112>
1670 IF TX$="{RVSON}KEINE{RVOFF}"THEN RETU
     RN                                       <052>
1680 IF AU$="BILDSCHIRM"THEN 1970             <052>
1690 PRINT"{2DOWN,2SPACE}ICH DRUCKE..."       <061>
1700 PRINT"{DOWN,SPACE}ZUM STOPPEN BITTE '
     ←' DRUECKEN."                            <048>
1710 OPEN 1,8,2,TX$:INPUT#255,FE              <045>
1720 IF FE<>0 THEN 1930                       <251>
1730 OPEN 2,4,1                               <200>
1740 PRINT#2,CHR$(27)"@";CHR$(27)CHR$(15); <143>
1750 PRINT#2,CHR$(27)CHR$(67)CHR$(0)CHR$(1
     2);                                      <126>
1760 IF MID$(FO$,1,1)="N"THEN ML=62:GOTO 1
     790                                      <050>
1770 PRINT#2,CHR$(27)CHR$(65)CHR$(6);         <246>
1780 PRINT#2,CHR$(27)CHR$(83)"0";:ML=128      <113>
1790 DF=INT((136-SP*ZS)/SP)                   <206>
1800 N=1:Z=1                                  <094>
1810 :                                        <008>
1820 PRINT#2,CHR$(27)CHR$(108)CHR$(N);        <234>
1830 SYS AD,ZS                                <192>
1840 IF PEEK(250)=222 THEN 1920               <039>
1850 GET A$:IF A$="←"THEN PRINT#2,"{2SPACE
     }*** BREAK ***":GOTO 1920                <192>
1860 Z=Z+1:IF Z<ML THEN 1830                  <221>
1870 N=N+DF+ZS:Z=1                            <090>
1880 IF N>136-ZS THEN 1900                    <117>
1890 PRINT#2,CHR$(27)CHR$(12);:GOTO 1820      <100>
1900 PRINT#2,CHR$(12);:GOTO 1800              <030>
1910 :                                        <108>
1920 CLOSE 1:CLOSE 2:RETURN                   <135>
1930 PRINT"{DOWN,WHITE,5SPACE}DISK-FEHLER.
     .. (RETURN)";                            <172>
1940 LA=0:GOSUB 2140:CLOSE 1:RETURN           <245>
1950 :                                        <148>
1960 :                                        <158>
1970 OPEN 1,8,2,TX$:INPUT#255,FE              <051>
1980 IF FE<>0 THEN 1930                       <001>
1990 PRINT"{CLR}";                            <150>
2000 SYS AD,39                                <147>
2010 GET A$:IF A$="←"THEN 2030                <219>
2020 IF PEEK(250)<>222 THEN 2000              <005>
2030 PRINT"{DOWN,2SPACE}** ENDE **"           <023>
2040 GET A$:IF A$=""THEN 2040                 <217>
2050 CLOSE 1:RETURN                           <145>
2060 :                                        <004>
2070 :                                        <014>
2080 :                                        <124>
2090 POKE 53280,14:POKE 53281,6:PRINT"{CLR
     ,LIG.BLUE}";                             <255>
2100 CLOSE 255:END                            <030>
2110 :                                        <054>
2120 :                                        <064>
2130 :                                        <074>
2140 T$="":L=0                                <099>
2150 PRINT"@";                                <073>
2160 GET A$:IF A$=CHR$(20)THEN 2220           <041>
2170 IF A$=CHR$(13)THEN PRINT CHR$(20):RET
     URN                                      <105>
2180 IF A$<CHR$(32)OR A$>"↑"THEN 2160         <020>
2190 IF L=LA THEN 2160                        <242>
2200 T$=T$+A$:L=L+1                           <246>
2210 PRINT"{LEFT}"A$;:GOTO 2150               <211>
2220 IF L=0 THEN 2160                         <220>
2230 L=L-1:T$=MID$(T$,1,L)                    <024>
2240 PRINT CHR$(20)CHR$(20);                  <053>
2250 GOTO 2150                                <086>
2260 :                                        <204>
2270 :                                        <214>
2280 :                                        <224>
2290 DATA 32,253,174,32,158,183,134,250,16
     2,1,32,198,255,166,250,32,207,255,157 <143>
2300 DATA 60,3,201,13,240,12,36,144,112,36
     ,202,208,239,169,13,157,60,3,32,204   <084>
2310 DATA 255,162,2,32,201,255,166,250,189
     ,60,3,32,210,255,189,60,3,202,201,13  <157>
2320 DATA 208,242,32,204,255,96,202,169,13
     ,157,60,3,169,222,133,250,208,240,0   <135>
```

Listing 1. »Multi-Lister« schafft Übersicht bei SMON- und Hypra-Ass-Ausdrucken.
Bitte mit dem Checksummer (Seite 159) eingeben.



Comic: Ingo Stein

Hypra-Comp bringt neuen Schwung

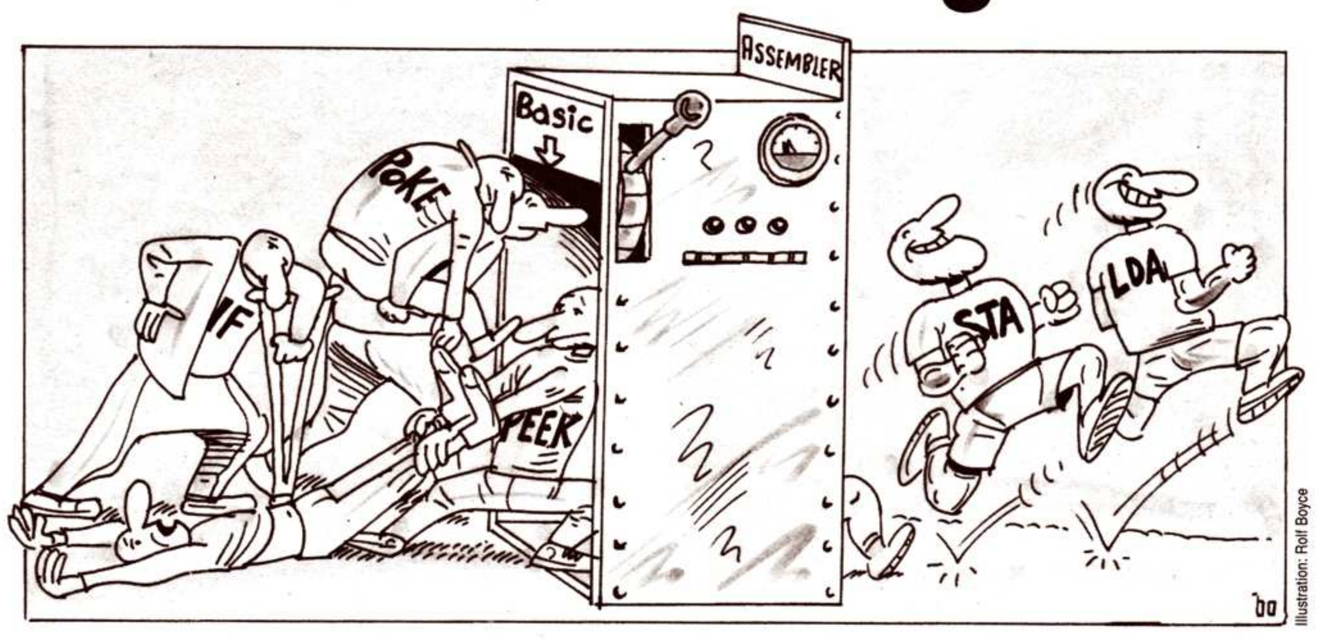


Illustration: Rolf Boyce

Zugegeben – das Basic 2.0 des C64 ist nicht die schnellste aller Basic-Versionen, was es für manche Anwendungen ungeeignet erscheinen läßt. Mit »Hypra-Comp« sind Sie dieser Probleme enthoben, da dieser Compiler aus Basic-Programmen schnellen Maschinen-Code erzeugt.

Hypra-Comp ist ein Basic-Compiler, der nahezu das gesamte Basic-Vokabular des C64 in Maschinensprache übersetzen kann. Da Hypra-Comp im Gegensatz zu manchen anderen Basic-Compilern reinen Maschinencode erzeugt, ist ein compiliertes Programm bis zu 10mal schneller als sein Quellprogramm. In Tabelle 1 sehen Sie einen Geschwindigkeitsvergleich zwischen einer compilierten und einer uncompilierten Version eines

Sortierte Elemente	Feld füllen normal	Feld füllen compiliert	Feld sortieren normal	Feld sortieren compiliert
10	0,22	0,1	2,15	0,32
50	0,82	0,48	13,6	2,8
100	1,6	0,97	29,0	5,6
500	8,1	4,8	172	33,8

Tabelle 1. Die Laufzeiten des uncompilierten und des compilierten Quicksort-Programms (Listing 3) in Sekunden

Quicksort-Algorithmus (Listing 1). Auch der Hilbert-Algorithmus (Listing 2), der eine Grafik wie in Bild 1 liefert, wird erstaunlich beschleunigt. Das Besondere an Hypra-Comp ist allerdings, daß er selbst fast vollständig in Basic geschrieben ist und sich vor der Benutzung erst einmal selbst compilieren muß.

Um mit Hypra-Comp arbeiten zu können, müssen Sie zunächst das Basic-Listing »Hypra-Comp« (Listing 3) mit dem Checksummer V 3.0 und den Maschinenspracheteil »RUNTIME« (Listing 4) mit dem MSE abtippen und auf eine ansonsten leere Diskette speichern. (Bitte beachten Sie hierbei die Eingabehinweise auf Seite 159). Es ist bereits jetzt möglich, Programme mit Hypra-Comp zu compilieren, doch wird die volle Geschwindigkeit beim Übersetzen erst dann erreicht, wenn sich das Programm selbst compiliert hat. Dazu laden Sie den Compiler mit

```
LOAD "HYPRA-COMP",8
```

von der Diskette und starten ihn mit RUN. Geben Sie auf die Frage nach dem Namen des Quelltextes »HYPRA-COMP« ein. Nun erfragt der Compiler die Längen einiger Variablen. Diese sind in der Reihenfolge, in der sie erfragt werden, in Tabelle 2 aufgeführt. Die ebenfalls in Tabelle 2 aufgeführten Bemerkungen über die einzelnen Variablen sind im Moment nicht weiter wichtig, und nur für denjenigen interessant, der den Compiler näher analysieren will.

Sobald sich der Compiler selbst übersetzt hat, was einige Zeit dauert, steht das compilierte Programm »C-HYPRA-COMP« zur Verfügung. Mt ihm können nun normale Basic-Programme in Maschinensprache übersetzt werden.

Bei der Übersetzung müssen Sie in zwei Schritten vorgehen:

1. Vorbereitung für eine Übersetzung

Da der Compiler nicht alle Befehle übersetzen kann, müssen zunächst folgende Befehle ersetzt oder entfernt werden:

RUN, SAVE, VERIFY, CONT, LIST, NEW, DEF

Da diese Befehle in der Regel nicht in Basic-Programmen vorkommen oder im Fall von DEFFN leicht zu ersetzen sind, sollte diese Hürde nicht schwer zu meistern sein.

Danach müssen Sie sich vergewissern, daß alle Feldvariablen mit DIM im Programm dimensioniert werden, und zwar in einer Zeile, die sich vor derjenigen befindet, in welcher die Feldvariable zum ersten Mal benutzt wird. Zu beachten ist, daß beim DIM-Befehl nur einfache Zahlen, jedoch keine Ausdrücke oder Variablen verwendet werden dürfen. (Zum Beispiel ist DIM A% (1, 3, 3) erlaubt, während DIM A (X) oder DIM A (3 * 5) zu einer Fehlermeldung führen würde.) Falls der Compiler auf eine Feldvariable stößt, ohne vorher eine dazugehörige DIM-Anweisung zu erreichen, fragt er den Benutzer nach der Dimensionierung.

Für seine Arbeit benötigt der Compiler die maximale Länge aller im Programm verwendeten Strings, die er bei der Compilation abfragt. Dem liegt das Prinzip zugrunde, daß im Compilat (das ist das übersetzte Programm) alle Strings feste Plätze im Speicher haben, wodurch die zeitaufwendige Garbage Collection entfällt. Falls Sie die maximale Stringlänge nicht ermitteln können oder wollen, geben Sie einfach den größtmöglichen Wert 255 an. Wenn das zu compilierende Programm jedoch sehr groß ist, oder wenn die Länge der Strings eines Stringfeldes angegeben werden soll, ist es ratsam, die maximale Länge genau zu bestimmen, um Speicherplatz zu sparen. Beispiel:

```
10 INPUT A$
20 KO$="TEST-TEST"
30 DIM C$(3)
40 FOR I=0 TO 3
50 READ C$(I)
60 PRINT A$,KO$,C$(I)
70 NEXT I
80 DATA DIES, SIND, DIE, DATEN
```

Erklärung: Die Variable A$ hat eine maximale Länge von 80 Zeichen, weil mit INPUT keine längeren Texte eingegeben werden können. Die Variable KO$ hat eine maximale Länge von neun Zeichen, da sie nach Zeile 20 nicht mehr verändert wird. Die Elemente des Feldes C$ (..) haben eine maximale Länge von fünf Zeichen, da sie über READ eingelesen werden und die Texte in der DATA-Zeile nicht länger als fünf Zeichen sind.

2. Compilation

Nach der Vorbereitung kann nun der Compiler an die Arbeit gehen. Dazu muß das zu compilierende Programm und das File »RUNTIME« auf einer Diskette stehen, auf der noch reichlich Platz ist (zirka das Dreifache der Länge des zu compilierenden Programms). Nun wird Hypra-Comp geladen und gestartet. Nach der Eingabe des Quellprogramm-Namens beginnt der Compiler mit Pass 1, wobei er die aktuelle Zeilennummer anzeigt. An den Stellen, wo er auf noch nicht benutzte Strings oder nicht dimensionierte Feldvariablen stößt, fragt der Compiler die oben erwähnten beschriebenen Werte ab. Beispiel:

```
10 DIM A (12, 4)
20 A (3, 0) = 1.45
30 B% (7) = 30000
40 A$ = "TEST-TEST"
50 DIM A$ (2, 2, 2)
60 A$ (1, 0, 0) = A$ + A$
```

Variablenname	Länge	(Bemerkung)
TR	5	Variable für Zahlen im MFLP-Format
C0$	1	CHR$(0)
GE$	1	CHR$(34)
NA$	20	Programmname
CH$	1	Variable für alle GET #
VA$( )	4	Feld für alle Variablennamen
I$	1	Hilfsvariable
J$	1	Hilfsvariable
ZL$	1	Zeilennummer Lo
ZH$	1	Zeilennummer Hi
KO$	80	Speicher für Stringkonstanten des Quellprogramms
HL$	2	Adresse im Hi/Lo-Format
N1$	1	Variablenname 1.Buchstabe
N2$	1	Variablenname 2.Buchstabe
N3$	1	Variablentyp
VA$	3	N1$ + N2$ + N3$
AR$( )	3	Adressen der Maschinenroutinen
PR$( )	1	Daten der Datei £S

Tabelle 2. Die beim Übersetzen von »Hypra-Comp« einzugebenden Variablenlängen mit ihrer Bedeutung

Abfrage des Compilers:		Eingabe:
Feld B%(...):	Anzahl der Dimensionen:	1
	Dimension Nr.1:	10
	(Dies wird nur abgefragt, weil das Feld B% ( ) nicht mit DIM dimensioniert wurde)	
Variable A$:	Länge:	9
	(Sofern A$ im weiteren Verlauf des Programms nicht größer werden kann.)	
Variable A$(...):	Länge:	18
	(Falls keinem Element des Feldes beim Programmlauf eine Zeichenkette zugewiesen wird, die länger als 18 Zeichen ist. (A$+A$= Zeile 60))	

Nach Pass 1 bearbeitet der Compiler das Programm ohne weitere Eingaben. Sollte er auf einen Fehler stoßen, so wird dieser gemeldet und die Compilation abgebrochen. Am Ende von Pass 1 gibt der Compiler den für Konstanten, Variablen und DATAs benötigten Speicherplatz an. Am Ende von Pass 2 folgt die Angabe der Länge des reinen Programms, sowie die Zeilenzahl des Quellprogramms. Falls keine Fehler auftraten, meldet sich Hypra-Comp nach Ausführung von Pass 3 mit READY. Das Compilat steht von nun ab unter »C-NAME« auf der Diskette und kann wie ein normales Basic-Programm geladen und gestartet werden.

Programmbeschreibung

Der Compiler arbeitet grob beschrieben so: Zuerst werden die Adressen aller Variablen festgelegt (Pass 1). Außerdem werden alle Konstanten und DATAs vom Programm getrennt in spezielle Datenfiles geschrieben. Danach werden in Pass 2 alle Befehle in Sprünge in die entsprechenden Unterprogramme umgesetzt, die im Maschinencode-Teil »RUNTIME« zu finden sind. Die Ausdrücke, die im Programm auftauchen, werden dabei so umgewandelt, daß sie im laufenden Programm schnell abgearbeitet werden können (UPN-Notation).

So arbeitet Hypra-Comp

Nach Pass 2 ist das Maschinenprogramm fast fertig, in Pass 3 werden nur noch Programmsprünge übersetzt, die zu höheren Zeilennummern verzweigen (z.B.: 10 GOTO 30). Schließlich erfolgt noch die Verbindung aller Einzelfiles (SYS-Zeile; Runtime; Konstante; DATAs und das Maschinenprogramm). Die erwähnten Dateien, die nach der Compilation gelöscht werden, also nur beim Compilationsabbruch auf der Diskette bleiben, haben folgende Namen:

»£S«

enthält die SYS-Zeile, die beim Auflisten des Compilats sichtbar wird, sowie eine Initialisierungsroutine für die Variablen im Compilat.

»£1«, »£2«, »£3«

enthalten das Programm in verschiedenen Entwicklungsstufen (Nach Pass 1, 2 und 3).

»£K«

enthält alle Konstanten des Quellprogramms.

»£D«

enthält alle DATAs des Quellprogramms.

»£X«

besteht aus den Dateien »£K« und »£D«

Hypra-Comp benutzt zur Beschleunigung der Arbeitsgeschwindigkeit des Compilats einige Techniken, die von der normalen Abarbeitung von Basic-Programmen abweichen und dadurch eventuell zu Schwierigkeiten beim Compilat führen können. Diese lassen sich jedoch umgehen. Die nachstehenden Erläuterungen sind Ihnen sicher eine gute Hilfe:

Schnelle Integer-Berechnungen

1. Integerarithmetik

Der Basic-Interpreter stellt zwar Integervariablen zur Verfügung, die im Speicher auch im Integerformat gespeichert werden; sämtliche Rechnungen werden aber im Fließkomma-Format durchgeführt.

Damit geht der Geschwindigkeitsvorteil, den Integervariablen normalerweise bieten, verloren. Hypra-Comp dagegen benutzt bei den Grundrechenarten, bei Vergleichen und bei logischen Operationen (AND, OR) spezielle Integerroutinen, falls beide Operanden Integerzahlen oder -variablen sind. Beispiel:

```
10 PRINT 3 *A% + B% - π
```

Das Compilat berechnet nun 3 * A% im Integer-Format, addiert im Integerformat B%, wandelt das Ergebnis ins Fließkomma-Format und subtrahiert davon den Wert π. Wegen dieser Eigenart des Compilers kann es bei compilierten Programmen zu folgenden Problemen kommen:

a) Überlauf bei Addition und Multiplikation

Beispiel: PRINT A% * 256. Das Compilat rechnet im Integerformat, da beide Argumente ganze Zahlen sind. Ist A% nun größer als 127, so wird das Ergebnis größer als 32767, es kann folglich nicht mehr im Integerformat dargestellt werden. Daher würde das Compilat im Gegensatz zum Quellprogramm, das im Fließkomma-Format rechnet, einen falschen Wert ausgeben.

b) Rundung bei Divisionen

Beispiel: PRINT 9 / 4. Das Compilat rechnet wiederum im Integerformat und gibt 2 statt 2.25 aus, weil bei der Integerrechnung nur ganze Zahlen auftreten.

Probleme treten also immer dann auf, wenn eine Rechnung mit zwei Integerzahlen/-variablen erfolgt, deren Ergebnis nicht als Integerzahl darstellbar ist. Man muß an den entsprechenden Stellen mindestens eines der beiden Argumente in eine Fließkommazahl umwandeln. Sie sollten also in Beispiel Nummer 1 entweder statt »A%« »A« benutzen oder an die Zahl 256 einen Punkt anhängen (256.). Dies beeinflußt den Interpreter nicht, aber der Compiler behandelt die Zahl jetzt als Fließkomma-Konstante. A% * 256. wird, weil nicht beide Argumente Intergerzahlen sind, im Fließkomma-Format gerechnet, und so wird das gewünschte Resultat erzielt.

2. GOSUB – RETURN

Aufgrund einer einfacheren und schnelleren GOSUB – RETURN-Bearbeitung sind abgebrochene FOR – NEXT-Schleifen nicht mehr möglich. Beispiel:

```
10 FOR A=1 TO 10 : GOSUB 30 : NEXT
20 END
30 FOR I=1 TO 20 : RETURN
```

Der Interpreter akzeptiert diesen Unsinn und bezieht das NEXT auf das FOR in Zeile 10 und nicht auf das in Zeile 30. Das Compilat dagegen schließt bei RETURN die noch offenen FOR – NEXT-Schleifen nicht. Es würde also das NEXT in Zeile 10 auf das FOR in Zeile 30 beziehen und sich an dieser Konstruktion »aufhängen«.

3. DATA-Bearbeitung

Der Compiler nimmt bei den DATAs eine Vorübersetzung vor und legt alle Zahlen im MFPT- oder Integerformat im Speicher ab. Dadurch können numerische DATAs nur noch in Zahlenvariablen hineingelesen werden, nicht aber in Stringvariablen. Beispiel:

```
10 READ A$
20 DATA 12345
```

Eine solche Konstruktion würde das Compilat mit einer Fehlermeldung quittieren. Um diese Klippe zu umschiffen, müssen einfach Anführungszeichen um die Zahl gesetzt werden: DATA "12345". "12345" wird vom Compiler als String erkannt und entsprechend gespeichert. Es kann in String-, aber nicht mehr in Zahlenvariablen eingelesen werden.

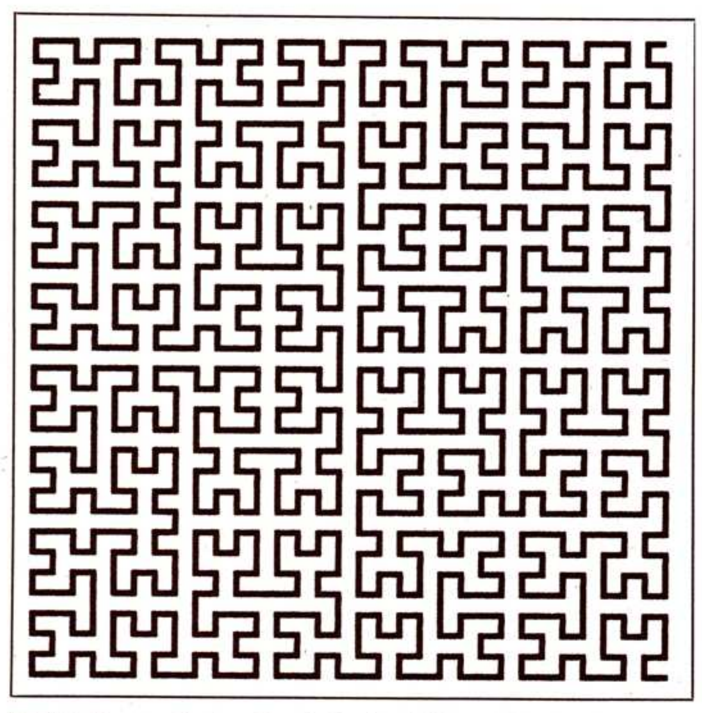


Bild 1. Das grafische Ergebnis des Hilbert-Algorithmus aus Listing 2 wird deutlich schneller erzeugt

4. Verändertes Input/ Input #

Beim Input-Befehl verhält sich das compilierte Programm etwas anders als das ursprüngliche Basic-Programm. Beim Einlesen von Zahlenvariablen wird kein »?REDO FROM START« ausgegeben, falls die Eingabe Buchstaben enthält, weil einer Zahlenvariablen immer der Wert VAL(Eingabe) zugewiesen wird. Bei allen Eingaben entfällt ein »?EXTRA IGNORED«, da alle eingegebenen Zeichen bis zum ⟨RETURN⟩ der Variablen zugewiesen werden. Anführungszeichen, Kommata und Doppelpunkt haben keine Sonderbedeutung mehr. Sie müssen beim Compilat alle Angaben, die mit einer Input-Anweisung gelesen werden sollen, durch ⟨RETURN⟩ trennen.

Beispiel:

```
10 INPUT A$, B, C%
20 PRINT A$ : PRINT B : PRINT C%
```

Eingabe: TEXT, 10.45, 1 ⟨RETURN⟩

Basic-Programm gibt aus:	Compilat gibt aus:
Text	Text,10.45,1
10.45	0
1	0

Dieses Problem kann man umgehen, indem man für jede Variable einen eigenen Input-Befehl benutzt.

5. LOAD-Befehl

Beim Nachladen von Programmen gibt es nur dann Probleme, wenn mit LOAD "Name",8 ein weiteres Basic-Programm gestartet werden soll. In diesem Fall muß das nachgeladene Programm ebenfalls mit Hypra-Comp compiliert sein, sonst stürzt das Programm ab.

6. Fehlerhafte Basic-Programme

Hypra-Comp entdeckt zwar syntaktische, aber keine logischen Fehler. Diese Fehler, die im Basic-Programm zu normalen Fehlermeldungen führen würden, können das Compilat zum Absturz bringen. Deshalb sollten die zu übersetzenden Programme gründlich ausgetestet sein.

(Armin Biernaczyk/sk)

```
100 DIM F%(1000),L%(50),R%(50)                    <181>
105 INPUT"ANZAHL DER ELEMENTE:";N%                 <098>
110 PRINT"FELD FUELLEN...{2SPACE}";:T=TI           <226>
130 FOR I=1 TO N%:F%(I)=RND(1)*32000:NEXT          <020>
140 T=TI-T:PRINT T/60"SEC"                         <012>
160 PRINT"FELD SORTIEREN... ";:T=TI                <065>
180 L%(1)=1:R%(1)=N%:T%=0:GOSUB 500                <230>
190 T=TI-T:PRINT T/60"SEC"                         <062>
220 GOTO 105                                       <052>
230 :                                              <206>
500 T%=T%+1:IF L%(T%)>=R%(T%)THEN 600              <176>
510 L%=L%(T%):R%=R%(T%):V%=F%((L%+R%)/2)           <188>
520 IF L%>R%THEN 580                               <188>
530 IF F%(L%)<V%THEN L%=L%+1:GOTO 530              <220>
540 IF F%(R%)>V%THEN R%=R%-1:GOTO 540              <175>
550 IF L%>R%THEN 580                               <218>
560 H%=F%(L%):F%(L%)=F%(R%):F%(R%)=H%              <095>
570 L%=L%+1:R%=R%-1:GOTO 520                       <049>
580 R%(T%+1)=R%:L%(T%+1)=L%(T%):GOSUB 500          <153>
590 L%(T%+1)=L%:R%(T%+1)=R%(T%):GOSUB 500          <223>
600 T%=T%-1:RETURN                                 <136>
```

Listing 1. Das »Quicksort«-Programm für den Geschwindigkeitsvergleich in Tabelle 1. Bitte mit dem Checksummer (Seite 159) eingeben.

```
1 DIM W%(7):DIM B%(7)                              <184>
10 INPUT"TIEFE (1-7)";O%                           <043>
20 IF O%<1 OR O%>7 THEN 10                         <236>
30 M%=2↑(7-O%)                                     <050>
90 :                                               <066>
100 :                                              <076>
101 : REM   HAUPTPROGRAMM                          <170>
102 :                                              <078>
120 GOSUB 10000:GOSUB 200:POKE 198,0:WAIT
    198,1:GOSUB 14000:GOTO 10                      <165>
130 :                                              <106>
160 :                                              <136>
165 : REM   HILBERT-PROZEDUR                       <144>
170 :                                              <146>
200 IF O%=0 THEN O%=O%+B%:RETURN                   <244>
205 W%=W%-W%(O%):O%=O%-B%:W%(O%)=-W%(O%+B%
    ):GOSUB 200                                    <233>
220 GOSUB 11400:W%=W%+W%(O%):O%=O%-B%:W%(O
    %)=W%(O%+B%):GOSUB 200                         <108>
260 GOSUB 11400:O%=O%-B%:W%(O%)=W%(O%+B%):
    GOSUB 200                                      <037>
280 W%=W%+W%(O%):GOSUB 11400:O%=O%-B%:W%(O
    %)=-W%(O%+B%):GOSUB 200                        <081>
290 W%=W%-W%(O%):O%=O%+B%:RETURN                   <026>
900 :                                              <114>
910 :                                              <124>
10000 :                                            <070>
10001 : REM  INITIALISIEREN                        <155>
10002 :                                            <072>
10005 POKE 56576,198:POKE 56578,63                 <056>
10020 POKE 53272,120:POKE 648,92                   <240>
10030 POKE 53265,187:I%=24576                      <066>
10040 POKE I%,0                                    <017>
10045 I%=I%+1:IF I%<=32575 GOTO 10040              <140>
10047 I%=23552                                     <075>
10050 POKE I%,16                                   <118>
10052 I%=I%+1:IF I%<=24551 GOTO 10050              <238>
10055 W%=1:A%=4:B%=1:Y%=170:X%=60                  <040>
10057 W%(O%)=1                                     <121>
10062 B%(7)=1:B%(6)=2:B%(5)=4:B%(4)=8:B%(3
      )=16:B%(2)=32:B%(1)=64:B%(0)=128             <163>
10070 RETURN                                       <222>
10075 :                                            <145>
10077 :                                            <147>
10078 : REM    PLOT                                <155>
10080 :                                            <150>
11000 J%=(X%AND 504)+40*(Y%AND 248)+(Y%AND
      7)+24576                                     <063>
11020 POKE J%,PEEK(J%)OR B%(X%AND 7):RETUR
      N                                            <133>
11400 :                                            <200>
11401 :                                            <201>
11402 : REM    MOVE                                <058>
11403 :                                            <203>
11410 I%=M%                                        <085>
11420 ON W%AND 3 GOTO 11520,11540,11560            <190>
11440 :                                            <240>
11500 X%=X%+1:GOSUB 11000                          <067>
11505 I%=I%-1:IF I%THEN 11500                      <165>
11510 RETURN                                       <136>
11520 Y%=Y%-1:GOSUB 11000                          <253>
11525 I%=I%-1:IF I%THEN 11520                      <191>
11530 RETURN                                       <158>
11540 X%=X%-1:GOSUB 11000                          <237>
11545 I%=I%-1:IF I%THEN 11540                      <215>
11550 RETURN                                       <178>
11560 Y%=Y%+1:GOSUB 11000                          <165>
11565 I%=I%-1:IF I%THEN 11560                      <239>
11570 RETURN                                       <198>
13140 :                                            <162>
13150 :                                            <172>
14000 POKE 53265,27:POKE 53272,21                  <243>
14005 POKE 648,4                                   <159>
14010 POKE 56578,63:POKE 56576,151                 <239>
14020 RETURN                                       <106>
```

Listing 2. Auch die Berechnung des Hilbert-Algorithmus bekommt Schwung. Bitte mit dem Checksummer (Seite 159) eingeben.

```
10 REM      HYPRA-COMP                             <188>
11 REM VON ARMIN BIERNACZYK                        <000>
12 REM AN DER PAPENBURG 41                         <148>
13 REM 4630 BOCHUM 6                               <028>
14 REM TEL. (0 23 27) 3 55 76                      <212>
15 :                                               <247>
100 GOSUB 2210:GOSUB 8010:END                      <077>
130 :                                              <106>
1001 :                                             <215>
1002 : REM --- PASS 1 INITIALISIERUNG              <000>
1003 :                                             <217>
1010 POKE 53280,14:POKE 53281,6:TR$="1234"
     +"5":C0$=CHR$(0):GE$=CHR$(34)                 <241>
1015 PRINT"{CLR,DOWN,LIG.BLUE,RVSON,4SPACE
     }**** BASIC V2 COMPILER V1.4 ****{4SP
     ACE}"                                         <225>
1025 INPUT"{2DOWN,5SPACE}PROGRAMMNAME :";N
     A$:OPEN 1,8,15,"S:£?":OPEN 2,8,2,NA$+
     ",P,R"                                        <173>
1030 INPUT#1,A:CLOSE 2:CLOSE 1:IF A=0 THEN
      1050                                         <252>
1032 PRINT" DISK ERROR <SPACE>"                    <176>
1040 GET CH$:ON(CH$=" ")+2 GOTO 1015,1040          <188>
1050 PRINT"{DOWN,YELLOW}PASS 1{LIG.BLUE}":
     OPEN 1,8,15,"UJ":FOR I=1 TO 3000:NEXT
     :CLOSE 1                                      <105>
1058 OPEN 5,8,5,"£D,P,W":PRINT#5:CLOSE 5           <233>
1060 OPEN 2,8,2,NA$+",P,R":GET#2,CH$,CH$:O
     PEN 4,8,4,"£K,P,W":OPEN 3,8,3,"£1,S,W
     "                                             <238>
1070 PRINT#4:DIM DI%(3),VA$(120),VA(120)           <093>
1075 DF%=0:VZ%=0:DZ%=0:VA=40960:AN=5030+1:
     AD=AN:TT%=0                                   <016>
1080 FOR I=828 TO 834:READ I%:POKE I,I%:NE
     XT:POKE 785,60:POKE 786,3:RETURN              <064>
1097 :                                             <057>
1098 : REM --- ZEILENNUMMER HOLEN                  <214>
1099 :                                             <059>
1210 GET#2,I$,J$:IF I$+J$=""THEN PRINT#3,"
     ↑↑";:FL%=0:RETURN                             <012>
1225 GOSUB 40600:PRINT ASC(ZL$)+ASC(ZH$)*2
     56."{UP}":PRINT#3,ZL$ZH$;:FL%=-1:RETU
     RN                                            <086>
1241 :                                             <201>
1242 : REM --- KONSTANTE SPEICHERN                 <114>
1243 :                                             <203>
1250 IF DF%THEN CLOSE 5:DF%=0:OPEN 4,8,4,"
     £K,P,A"                                       <170>
```

Listing 3. »Hypra-Comp«-Hauptprogramm. Bitte mit dem Checksummer (Seite 159) eingeben.

```
1252 ON 2-KT%GOTO 1270,1275:PRINT#4,CHR$(L
     EN(KO$))KO$;                              <147>
1255 I=AD:GOSUB 40010:PRINT#3,CHR$(KT%)HL$
     ;:AD=AD+LEN(KO$)-(KT%=2):RETURN           <208>
1270 IF KO>255 THEN I=KO:GOSUB 40010:KO$=H
     L$:PRINT#4,KO$;:GOTO 1255                 <254>
1272 PRINT#3,"{CTRL-G}"CHR$(KO);:RETURN        <095>
1275 I=USR(KO):KO$=TR$:PRINT#4,KO$;:GOTO 1
     255                                       <239>
1301 :                                         <007>
1302 : REM --- KONSTANTE EINLESEN              <001>
1303 :                                         <009>
1310 KO$="":KT%=2:I%=CH$<>GE$:IF I%THEN KO
     $=CH$                                     <167>
1315 GET#2,CH$:IF CH$=""THEN RETURN            <184>
1320 IF CH$<>GE$AND(CH$<>","OR NOT I%)THEN
     KO$=KO$+CH$:GOTO 1315                     <040>
1325 IF CH$<>GE$THEN RETURN                    <209>
1330 GOTO 2090                                 <041>
1341 :                                         <047>
1350 KO$=CH$:KT%=-1                            <020>
1355 GOSUB 2090:IF CH$<"0"OR CH$>"9"THEN 1
     365                                       <109>
1360 KO$=KO$+CH$:GOTO 1355                     <206>
1365 IF CH$="E"OR CH$="."THEN KT%=0:GOTO 1
     360                                       <192>
1370 IF(CH$="ō"OR CH$="N̄")AND RIGHT$(KO$,1
     )="E"THEN 1360                            <218>
1375 KO=VAL(KO$):KT%=-(KO=INT(KO)AND ABS(K
     O)<32768 AND KT%):RETURN                  <181>
1500 :                                         <206>
1501 : REM --- BENUTZERABFRAGEN                <087>
1502 :                                         <208>
1510 PRINT N1$N2$;:IF(ASC(N3$)AND 3)=1 THE
     N PRINT"%";:GOTO 1520                     <246>
1515 IF(ASC(N3$)AND 3)=2 THEN PRINT"$";        <070>
1520 IF ASC(N3$)>3 THEN PRINT" (...)";         <012>
1525 PRINT:RETURN                              <030>
1527 :                                         <233>
1530 PRINT"{2DOWN,3SPACE}VARIABLE: ";:GOSU
     B 1510                                    <220>
1535 INPUT"{DOWN,3SPACE}MAX. LAENGE";LE:LE
     =LE+1                                     <222>
1540 PRINT"{3UP}";:FOR I=1 TO 3:PRINT"{35S
     PACE}":NEXT                               <082>
1545 PRINT"{5UP}";:RETURN                      <125>
1601 :                                         <053>
1610 PRINT"{2DOWN,3SPACE}FELD: ";:GOSUB 15
     10:DI%(3)=0:DI%(2)=0:DI%(1)=0             <061>
1615 INPUT"{DOWN,3SPACE}WIEVIELE DIMENSION
     EN";DI%:IF DI%>3 OR DI%<1 THEN PRINT"
     {2UP}";:GOTO 1615                         <198>
1620 FOR I=1 TO DI%:PRINT"{3SPACE}DIMENSIO
     N NR."I"{3SPACE}:";:INPUT DI%(I)          <240>
1625 IF DI%(I)<1 OR DI%(I)>32000 THEN PRIN
     T"{UP}";:I=I-1                            <169>
1630 NEXT:PRINT"{3SPACE}OK ? (J/N)"            <092>
1635 GET I$:IF I$=""THEN 1635                  <107>
1640 PRINT LEFT$("{7UP}",DI%+4);               <249>
1645 FOR I=1 TO 7:PRINT"{34SPACE}":NEXT:PR
     INT"{9UP}";                               <212>
1650 IF I$<>"J"THEN 1610                       <117>
1655 VA=VA-(DI%(1)+1)*(DI%(2)+1)*(DI%(3)+1
     )*LE:IF VA<=AD THEN 60070                 <185>
1660 I=VA:GOSUB 40010:PRINT#4,HL$CHR$(DI%*
     2+1);                                     <111>
1665 DI%(3)=DI%(1)*DI%(2)*LE:DI%(2)=DI%(1)
     *LE:DI%(1)=LE:FOR J=1 TO DI%:I=DI%(J)     <252>
1670 GOSUB 40010:PRINT#4,HL$;:NEXT:AD=DI%*
     2+3+AD:RETURN                             <145>
1701 :                                         <153>
1702 : REM --- VARIABLENNAME LESEN             <155>
1703 :                                         <155>
1710 N1$=CH$:GET#2,CH$:N2$=CH$:I%=-1           <155>
1715 IF(CH$>"9"OR CH$<"0")AND(CH$>"Z"OR CH
     $<"A")GOTO 1725                           <044>
1720 GET#2,CH$:I%=0:GOTO 1715                  <116>
1725 IF I%THEN N2$=C0$                         <140>
1730 N3$=CHR$(-(CH$="%")-2*(CH$="$")):IF N
     3$=C0$THEN 1740                           <221>
1735 GET#2,CH$                                 <050>
1740 IF CH$=" "THEN 1735                       <172>
1745 IF CH$="("THEN N3$=CHR$(ASC(N3$)+4):G
     OSUB 2090                                 <050>
1750 VA$=N1$+N2$+N3$:RETURN                    <224>
1801 :                                         <255>
1802 : REM --- VARIABLENADRESSE HOLEN          <096>
1803 :                                         <001>
1810 GOSUB 1710:I%=1                           <135>
1811 IF VA$=LEFT$(VA$(I%),3)THEN 1820          <092>
1812 I%=I%+1:IF I%<=VZ%THEN 1811               <222>
1815 GOTO 1835                                 <010>
1820 I=VA(I%):GOSUB 40010:PRINT#3,MID$(VA$
     (I%),3,1)HL$;                             <222>
1825 IF LEN(VA$(I%))=4 THEN PRINT#3,RIGHT$
     (VA$(I%),1);                              <152>
1830 RETURN                                    <110>
1835 IF VA$="ST"+C0$THEN PRINT#3,"{CTRL-C,
     CTRL-A}"CHR$(111);:RETURN                 <123>
1840 IF VA$="TI"+C0$THEN PRINT#3,"{CTRL-C}
     "C0$CHR$(112);:RETURN                     <218>
1845 IF VA$="TI{CTRL-B}"THEN PRINT#3,"{CTR
     L-C,CTRL-B}"CHR$(113);:RETURN             <182>
1850 VZ%=VZ%+1:VA$(VZ%)=VA$:IF N3$=C0$THEN
     VA=VA-5:VA(VZ%)=VA:I%=VZ%:GOTO 1820       <202>
1855 IF N3$="{CTRL-A}"THEN VA=VA-2:VA(VZ%)
     =VA:I%=VZ%:GOTO 1820                      <009>
1860 IF N3$="{CTRL-B}"THEN GOSUB 1530:VA=V
     A-LE:VA(VZ%)=VA:I%=VZ%:GOTO 1820          <227>
1865 IF N3$<"{CTRL-F}"THEN LE=-(N3$="{CTRL
     -D}")*5-(N3$="{WHITE}")*2:GOTO 1875       <106>
1870 GOSUB 1530                                <224>
1875 GOSUB 1610:VA(VZ%)=AD-3-2*DI%:I%=VZ%:
     VA$(VZ%)=VA$(VZ%)+CHR$(DI%):GOTO 1820     <117>
1900 :                                         <098>
1902 : REM --- DATAZEILENBEARBEITUNG           <122>
1903 :                                         <101>
1910 IF NOT DF%THEN CLOSE 4:DF%=-1:OPEN 5,
     8,5,"£D,P,A"                              <144>
1912 GOSUB 2090                                <131>
1915 IF CH$=""OR CH$=":"THEN DZ%=DZ%+3:PRI
     NT#5,"{CTRL-A}"C0$C0$;:RETURN             <197>
1920 IF CH$=","THEN DZ%=DZ%+3:PRINT#5,"{CT
     RL-A}"C0$C0$;:GOTO 1912                   <132>
1925 IF(CH$>"9"OR CH$<"0")AND CH$<>"+"AND
     CH$<>"-"THEN 1940                         <021>
1927 IF CH$="+"THEN CH$="N̄"                    <124>
1928 IF CH$="-"THEN CH$="ō"                    <128>
1930 GOSUB 1350:ON KT%GOTO 1932:DZ%=DZ%+6:
     I=USR(KO):PRINT#5,C0$TR$;:GOTO 1945       <199>
1932 IF KO<0 THEN KO=65536+KO                  <044>
1935 DZ%=DZ%+3:I=KO:GOSUB 40010:PRINT#5,"{
     CTRL-A}"HL$;:GOTO 1945                    <214>
1940 GOSUB 1310:I%=LEN(KO$):DZ%=DZ%+I%+2:P
     RINT#5,"{CTRL-B}"CHR$(I%)KO$;             <218>
1945 IF CH$=""OR CH$=":"THEN RETURN            <110>
1950 IF CH$=","THEN 1912                       <149>
1955 GOTO 60030                                <117>
2000 :                                         <198>
2001 : REM --- TOKENVORBEARBEITUNG             <216>
2002 :                                         <200>
2010 CH%=ASC(CH$):IF CH%=131 THEN 1910         <191>
2012 IF CH%=134 GOTO 2104                      <022>
2020 IF CH%=136 THEN 2090                      <170>
2022 IF CH%<>137 THEN 2032                     <230>
2023 IF TT%THEN PRINT#3,"N̄";:TT%=0             <093>
2024 PRINT#3,CH$;:GOSUB 2090:IF CH$=","THE
     N I=0:GOSUB 40200:GOTO 2024               <061>
2026 GOSUB 1350:I=KO:GOSUB 40200:IF CH$=",
     "THEN 2024                                <190>
2028 RETURN                                    <052>
2032 IF CH%=139 THEN TT%=-1:GOTO 2084          <116>
2034 IF CH%=141 THEN 2024                      <207>
2035 IF CH%<>143 THEN 2038                     <232>
2036 GET#2,CH$:IF CH$<>""GOTO 2036             <252>
2037 RETURN                                    <061>
2038 IF CH%<>161 THEN 2050                     <100>
2040 GOSUB 2090:IF CH$="#"THEN PRINT#3,CHR
     $(162);:GOTO 2090                         <159>
2042 PRINT#3,CHR$(161);:RETURN                 <232>
2050 IF CH%<>167 THEN 2060                     <202>
2052 PRINT#3,CH$;:TT%=0:GOSUB 2090             <077>
2054 IF CH$>="0"AND CH$<="9"THEN PRINT#3,C
     HR$(137);:GOSUB 1350:I=KO:GOTO 40200      <181>
2056 RETURN                                    <082>
2060 IF CH%<180 OR CH%>202 THEN 2070           <233>
2062 PRINT#3,CH$;                              <162>
```

```
2064 GOSUB 2090:IF CH$="("THEN 2090               <174>
2066 GOTO 60040                                   <039>
2070 IF CH%=255 THEN PRINT#3,C0$"££S";:GOTO
      2090                                        <077>
2080 I%=CH%=138 OR CH%=148 OR CH%=149 OR C
     H%=150 OR CH%=154 OR CH%=155 OR CH%=1
     62 OR CH%=203                                <195>
2082 IF I%THEN 60050                              <239>
2084 PRINT#3,CH$;                                 <184>
2090 GET#2,CH$:IF CH$=" "THEN 2090                <109>
2095 RETURN                                       <121>
2100 :                                            <044>
2101 : REM --- DIM-ABFRAGE                        <031>
2102 :                                            <046>
2104 DI%=0:DI%(1)=0:DI%(2)=0:DI%(3)=0:GOSU
     B 2090:GOSUB 1710:IF N3$<"{CTRL-D}"GO
     TO 60090                                     <112>
2106 GOSUB 1350:DI%=DI%+1:IF KO<0 OR KO>30
     000 GOTO 60020                               <228>
2108 DI%(DI%)=KO:IF DI%<=3 AND CH$=","THEN
      GOSUB 2090:GOTO 2106                        <078>
2110 I%=1:IF CH$<>")"GOTO 60020                   <089>
2112 IF VA$=LEFT$(VA$(I%),3)THEN 2122             <187>
2114 I%=I%+1:IF I%<=VZ%THEN 2112                  <230>
2116 IF N3$<"{CTRL-F}"THEN LE=-(N3$="{CTRL
     -D}")*5-(N3$="{WHITE}")*2:GOTO 2120          <030>
2118 GOSUB 1530                                   <218>
2120 VZ%=VZ%+1:VA$(VZ%)=VA$+CHR$(DI%):VA(V
     Z%)=AD:GOSUB 1655                            <113>
2122 GOSUB 2090:IF CH$=","GOTO 2104               <199>
2124 RETURN                                       <150>
2200 :                                            <144>
2201 : REM ***** PASS 1 HAUPTPROGRAMM             <233>
2202 :                                            <146>
2210 GOSUB 1010                                   <166>
2215 GOSUB 1210:IF NOT FL%THEN 2300               <058>
2220 GOSUB 2090                                   <185>
2225 IF CH$=""THEN PRINT#3,CHR$(13);:GOTO
     2215                                         <015>
2230 IF CH$>CHR$(127)THEN GOSUB 2010:GOTO
     2225                                         <077>
2235 IF CH$=GE$THEN GOSUB 1310:GOSUB 1250:
     GOTO 2225                                    <058>
2240 IF(CH$>="0"AND CH$<="9")OR CH$="."THE
     N GOSUB 1350:GOSUB 1250:GOTO 2225            <100>
2245 IF CH$="("OR CH$=")"OR CH$=";"OR CH$=
     ":"OR CH$=","THEN PRINT#3,CH$;:GOTO 2
     220                                          <149>
2290 IF CH$>="A"AND CH$<="Z"THEN GOSUB 181
     0:GOTO 2225                                  <054>
2295 GOTO 60060                                   <140>
2300 PR=DZ%+2+AD:P2=PR:CLOSE 4:CLOSE 3:CLO
     SE 2:CLOSE 5                                 <204>
2305 OPEN 5,8,5,"£D,P,A":PRINT#5,"↑";:CLOS
     E 5:OPEN 1,8,15,"C:£X=£K,£D":CLOSE 1         <144>
2315 PRINT"{DOWN}VARIABLEN :"40960-VA" BYT
     ES"                                          <163>
2320 PRINT"KONSTANTEN:"AD-AN" BYTES"              <000>
2325 PRINT"DATAZEILEN:"DZ%" BYTES{DOWN}"          <134>
2330 IF VA<=PR THEN 60070                         <089>
2335 RETURN                                       <107>
3000 :                                            <182>
3001 : REM --- PASS 2 INITIALISIERUNG             <231>
3002 :                                            <184>
3010 DIM AR$(142),S1%(20,6),S2%(20),FU%(22
     ,1),PR%(6),PR$(68)                           <208>
3050 FOR I=0 TO 142:READ I1%,I2%:AR$(I)="
     "+CHR$(I1%)+CHR$(I2%):NEXT                   <135>
3060 FOR I=0 TO 22:READ FU%(I,0),FU%(I,1):
     NEXT:FOR I=0 TO 6:READ PR%(I):NEXT           <154>
3070 FOR I=0 TO 68:READ I%:PR$(I)=CHR$(I%)
     :NEXT                                        <233>
3072 I=PR:GOSUB 40010:PR$(19)=CHR$(L%):PR$
     (20)=CHR$(H%)                                <087>
3076 I=VA:GOSUB 40010:PR$(27)=CHR$(L%):PR$
     (22)=CHR$(H%)                                <145>
3080 I=AD+1:GOSUB 40010:PR$(61)=CHR$(L%):P
     R$(65)=CHR$(H%)                              <064>
3084 PR$(53)=PR$(27):PR$(57)=PR$(22)              <008>
3090 OPEN 4,8,4,"£S,P,W":FOR I=0 TO 68:PRI
     NT#4,PR$(I);:NEXT:CLOSE 4                    <236>
3095 OPEN 4,8,4,"£2,P,W":OPEN 2,8,2,"£1,S,
     R":PRINT"{DOWN,YELLOW}PASS 2{LIG.BLUE
     }":RETURN                                    <210>
3200 :                                            <128>
3201 : REM ***** AUSDRUCKSAUSWERTUNG              <076>
3202 :                                            <130>
3210 I%=S1%(S1%,1):IF I%>12 THEN 3255             <246>
3215 IF S2%<2 THEN 60020                          <184>
3220 IF(S2%(S2%)OR S2%(S2%-1))<2 GOTO 3235 <067>
3222 IF S2%(S2%)<2 OR S2%(S2%-1)<2 GOTO 32
     30                                           <056>
3225 IF I%=0 THEN S2%=S2%-1:PRINT#4,AR$(0)
     ;:GOTO 3245                                  <189>
3227 IF I%>6 AND I%<13 THEN PRINT#4,AR$(10
     7+I%);:GOTO 3240                             <055>
3230 GOTO 60090                                   <252>
3235 PRINT#4,AR$(S2%(S2%)+2*S2%(S2%-1)+4*I
     %+1);                                        <107>
3240 S2%=S2%-1:S2%(S2%)=-(S2%(S2%)=1 AND S
     2%(S2%+1)=1 AND I%<>4)OR-(I%>6)              <113>
3245 GOSUB 40700:S1%=S1%-1:RETURN                 <082>
3255 IF I%>14 THEN 3267                           <046>
3257 IF S2%<1 THEN GOTO 60020                     <076>
3260 IF S2%(S2%)<2 THEN PRINT#4,AR$(2*(I%-
     13)+53+S2%(S2%));:GOTO 3245                  <164>
3265 GOTO 60090                                   <031>
3267 J1%=S1%(S1%,2):J2%=S2%(S2%)                  <166>
3269 IF I%>38 THEN 3310                           <042>
3270 IF S2%<(1-(I%>35)-(I%=38))THEN 60020         <062>
3272 IF(I%=32 OR I%=35)AND SR%THEN SR%=0:P
     RINT#4,AR$(95);:GOSUB 40700                  <049>
3275 IF J1%=J2%THEN 3300                          <227>
3280 IF J1%=3 AND(J2%=0 OR J2%=1)THEN 3300 <013>
3285 IF J1%=1 AND J2%=0 THEN PRINT#4,AR$(1
     03);:GOSUB 40700:S2%(S2%)=1:GOTO 3300 <000>
3290 IF J1%=0 AND J2%=1 THEN PRINT#4,AR$(1
     04);:GOSUB 40700:S2%(S2%)=0:GOTO 3300 <150>
3295 IF J1%<>4 THEN GOTO 60090                    <187>
3300 IF I%=15 OR I%=17 AND J2%=1 THEN 3302 <059>
3301 PRINT#4,AR$(2*(I%-16)+57-(S2%(S2%)=1)
     );:GOSUB 40700                               <047>
3302 S2%=S2%+(I%>35)+(I%=38)                      <228>
3305 IF S1%(S1%,3)<3 THEN S2%(S2%)=S1%(S1%
     ,3)                                          <199>
3306 S1%=S1%-1:RETURN                             <229>
3310 IF J2%=2 THEN GOTO 60090                     <091>
3315 IF J2%=0 THEN PRINT#4,AR$(103);:GOSUB
      40700                                       <150>
3320 PRINT#4,"£"CHR$(S1%(S1%,4))"{SHIFT-SP
     ACE}"CHR$(S1%(S1%,5));:GOSUB 40720           <132>
3325 S2%=S2%-S1%(S1%,6)+1:S2%(S2%)=S1%(S1%
     ,3)                                          <043>
3330 IF FL%AND S1%=1 THEN PRINT#4,AR$(99);
     :GOTO 3245                                   <224>
3335 PRINT#4,AR$(I%+66);:GOTO 3245                <250>
3500 :                                            <174>
3501 REM ***** VARIABLE AUF DEN STACK             <141>
3502 :                                            <176>
3510 IF I%>2 THEN 3550                            <033>
3520 GOSUB 40100:I$=CH$:GOSUB 40100:PRINT#
     4,"£"I$"{SHIFT-SPACE}"CH$AR$(I%+108);
     :GOSUB 40710                                 <173>
3525 S2%=S2%+1:S2%(S2%)=I%:RETURN                 <021>
3550 IF I%>3 THEN 3570                            <145>
3555 S2%=S2%+1:GOSUB 40100:S2%(S2%)=ASC(CH
     $):GOSUB 40100:GOSUB 40700                   <038>
3556 IF CH$<>CHR$(112)THEN 3560                   <161>
3557 IF SR%THEN PRINT#4,AR$(95);:GOSUB 407
     00                                           <104>
3558 SR%=0:PRINT#4,"8";:PR=PR+1.                  <064>
3560 PRINT#4,AR$(ASC(CH$));:RETURN                <211>
3570 IF I%=7 THEN GOSUB 40100:PRINT#4,"£"C
     H$AR$(65);:PR=PR+5.:I%=1:GOTO 3525           <142>
3580 S1%=S1%+1:S1%(S1%,0)=1:S1%(S1%,1)=I%+
     35:S1%(S1%,2)=1:S1%(S1%,3)=I%-4              <058>
3585 KL%=KL%+1:GOSUB 40100:S1%(S1%,4)=ASC(
     CH$)                                         <254>
3587 GOSUB 40100:S1%(S1%,5)=ASC(CH$)              <132>
3590 GET#2,CH$:S1%(S1%,6)=ASC(CH$):ON 3-AS
     C(CH$)GOTO 3600,3605                         <212>
3595 GOSUB 40800                                  <044>
3600 GOSUB 40800                                  <049>
```

Listing 3. (Fortsetzung)

```
3605 RETURN                                      <107>
3710 S1%=S1%+1:S1%(S1%,0)=1:S1%(S1%,1)=I%:
     S1%(S1%,2)=J1%:S1%(S1%,3)=J2%               <188>
3715 KL%=KL%+1:IF I%<36 THEN RETURN              <145>
3720 ON I%-36 GOTO 3725,3735                     <026>
3725 GOSUB 40800:S1%(S1%,2)=2:S1%(S1%,3)=2
     :RETURN                                     <216>
3735 GOSUB 40800:GOTO 3725                       <083>
3910 I%=-1:IF CH$<"E"AND CH$>"£"THEN I%=AS
     C(CH$)-170:GOTO 3975                        <196>
3920 IF CH$>"W"OR CH$<"E"THEN RETURN             <026>
3925 PR%=5:I$=CH$:GOSUB 40100                    <005>
3930 IF CH$>"W"OR CH$<"E"THEN 3955               <207>
3935 IF CH$=I$THEN 60020                         <062>
3940 IF CH$<>"R"AND I$<>"R"THEN I%=12:GOTO
      3975                                       <241>
3945 IF CH$="E"OR I$="E"THEN I%=10:GOTO 39
     75                                          <165>
3950 I%=11:GOTO 3975                             <115>
3955 IF I$="E"THEN I%=7:RETURN                   <166>
3965 IF I$="R"THEN I%=9:RETURN                   <114>
3970 I%=8:RETURN                                 <245>
3975 IF I%<7 THEN PR%=PR%(I%)                    <133>
3980 GOTO 40100                                  <190>
4010 IF PR%>S1%(S1%,0)THEN S1%=S1%+1:S1%(S
     1%,0)=PR%:S1%(S1%,1)=I%:RETURN              <019>
4015 IF S1%=0 THEN RETURN                        <212>
4020 RZ%=I%:GOSUB 3210:I%=RZ%:IF S1%(S1%+1
     ,0)>1 THEN 4010                             <103>
4025 RETURN                                      <017>
4500 :                                           <158>
4501 REM *****  AUSDRUCK HAUPTPROGRAMM           <195>
4502 :                                           <160>
4510 FL%=0:FU%=-1:RE%=0:S1%=0:S2%=0:KO%=0:
     KL%=0:SR%=-1                                <027>
4515 GOSUB 40100                                 <233>
4520 IF CH$>"J"OR CH$<"H"THEN 4550               <194>
4525 IF NOT FU%THEN GOTO 4685                    <073>
4530 RE%=0:I%=ASC(CH$)-164:J1%=FU%(I%-16,0
     ):J2%=FU%(I%-16,1):GOSUB 3710:GOTO 45
     15                                          <243>
4550 IF NOT RE%THEN 4600                         <037>
4555 GOSUB 3910:IF I%=-1 THEN 4610               <031>
4560 FU%=-1:RE%=0:GOSUB 4010:GOTO 4520           <102>
4600 IF CH$="N"THEN 4515                         <026>
4605 IF CH$="O"THEN S1%=S1%+1:S1%(S1%,0)=9
     :S1%(S1%,1)=13:GOTO 4515                    <201>
4610 IF CH$="Z"THEN S1%=S1%+1:S1%(S1%,0)=6
     :S1%(S1%,1)=14:RE%=0:GOTO 4515              <160>
4615 IF CH$<>"("THEN 4630                        <099>
4617 IF NOT FU%THEN 4685                         <243>
4620 RE%=0:KL%=KL%+1:S1%=S1%+1:S1%(S1%,0)=
     1:S1%(S1%,1)=15:S1%(S1%,2)=4                <003>
4625 S1%(S1%,3)=3:GOTO 4515                      <078>
4630 IF CH$<>")"THEN 4650                        <245>
4635 IF KL%=0 THEN 4685                          <238>
4640 KL%=KL%-1:FU%=0:RE%=-1:PR%=1:GOSUB 40
     10:IF S1%>0 OR NOT FL%GOTO 4515             <120>
4645 RETURN                                      <131>
4650 IF CH$<>","THEN 4670                        <143>
4655 IF KO%=0 THEN 4685                          <050>
4660 KO%=KO%-1:FU%=-1:RE%=0:PR%=1:GOSUB 40
     10:GOTO 4515                                <004>
4670 IF CH$>"{CTRL-G}"OR NOT FU%THEN 4685        <212>
4675 FU%=CH$>"{CTRL-C}"AND CH$<"{CTRL-G}":
     RE%=NOT FU%:I%=ASC(CH$):GOSUB 3510:GO
     TO 4515                                     <107>
4680 :                                           <084>
4681 REM --- AUSDRUCK ENDAUSWERTUNG              <006>
4682 :                                           <086>
4685 IF KO%<>0 OR KL%<>0 THEN 60100              <038>
4690 PR%=0:GOSUB 4010:IF S2%=1 THEN RETURN       <071>
4692 GOTO 60020                                  <252>
4700 :                                           <104>
4701 REM --- ZEILENNUMMER HOLEN                  <073>
4702 :                                           <106>
4710 GOSUB 40600                                 <078>
4720 IF ZH$="±"THEN FL%=-1:RETURN                <052>
4722 GOSUB 40900:ZN=L%+256.*H%:FL%=0:PRINT
      ZN"{UP}":RETURN                            <051>
4801 :                                           <205>
4802 : REM --- AUSDRUCKSTYP PRUEFEN              <111>
4803 :                                           <207>
4810 GOSUB 4510:IF S2%(1)=1 THEN RETURN          <142>
4815 IF S2%(1)=2 THEN 60090                      <248>
4820 PRINT#4,AR$(103);:GOTO 40700                <173>
4910 GOSUB 4510:IF S2%(1)=0 THEN RETURN          <243>
4915 IF S2%(1)=2 THEN GOTO 60090                 <056>
4920 PRINT#4,AR$(104);:GOTO 40700                <027>
4960 GOSUB 4510:IF S2%(1)=2 THEN RETURN          <039>
4965 GOTO 60090                                  <209>
5001 :                                           <151>
5002 : REM --- LET-BEFEHL, LINKE SEITE           <248>
5003 :                                           <153>
5010 PF%=0                                       <223>
5012 LE%=ASC(CH$)AND 3:PH%=-1                    <016>
5015 IF CH$>"{CTRL-B}"THEN 5045                  <237>
5020 GOSUB 40100:I$=CH$:GOSUB 40100              <129>
5025 IF PF%THEN PRINT#4,"£"I$"{SHIFT-SPACE
     }"CH$;:GOSUB 40720:GOTO 40100               <068>
5030 PL%=ASC(I$):PH%=ASC(CH$):GOTO 40100         <140>
5045 IF CH$="{CTRL-C}"THEN 5070                  <107>
5050 FL%=-1:FU%=-1:RE%=0:S1%=0:S2%=0:KO%=0
     :KL%=0:SR%=-1                               <245>
5055 I%=ASC(CH$):GOSUB 3580:GOSUB 4515           <206>
5057 IF PF%THEN PRINT#4,CHR$(165)CHR$(71)C
     HR$(164)CHR$(72);:GOSUB 40720:GOTO 40
     100                                         <000>
5060 PRINT#4,"GGHGHH";                           <126>
5065 PR=PR+6.:GOTO 40100                         <131>
5070 GET#2,CH$:IF CH$="{CTRL-B}"THEN GET#2
     ,CH$,CH$:RETURN                             <032>
5075 GOTO 60120                                  <157>
5101 :                                           <251>
5102 : REM ***** VARIABLENZUWEISUNG              <247>
5103 :                                           <253>
5110 GOSUB 5010:IF CH$<>"R"THEN 60020            <017>
5115 IF LE%=3 THEN 5135                          <175>
5120 ON LE%+1 GOSUB 4910,4810,4960               <100>
5122 IF PH%=-1 THEN PRINT#4,CHR$(104)CHR$(
     168)CHR$(104);:GOSUB 40700:GOTO 5130        <025>
5125 PRINT#4,"£"CHR$(PL%)"{SHIFT-SPACE}"CH
     R$(PH%);:GOSUB 40720                        <007>
5130 PRINT#4,AR$(LE%+120+55*(LE%=2));:GOTO
      40700                                      <202>
5135 GOSUB 4510:IF S2%(1)<>2 THEN 60090          <112>
5140 PRINT#4," S"CHR$(14)"I{CTRL-F}"CHR$(2
     40)"{CTRL-C}LHRGF8"CHR$(233)"{CTRL-F,
     F1}"CHR$(34)"{SHIFT-SPACE}O";               <065>
5150 PRINT#4,CHR$(132)"#"AR$(90)" "CHR$(23
     1)"£";                                      <103>
5160 PR=PR+27.:RETURN                            <082>
5201 :                                           <097>
5202 : REM ***** TOKENAUSWERTUNG                 <177>
5203 :                                           <099>
5210 CH%=ASC(CH$):IF CH%>156 THEN ON CH%-1
     56 GOTO 6665,6715,6765,6865,6915,6960       <108>
5212 IF CH%>143 THEN ON CH%-143 GOTO 6120,
     6162,6265,6320,,,,6415,6465,6512,,,66
     15                                          <211>
5213 ON CH%-128 GOTO 5315,5415,,5515,5615,
     ,5715,5815,5862,,5915,5965,6015,6065        <087>
5215 :                                           <111>
5216 : REM --- END                               <038>
5217 :                                           <113>
5250 PRINT#4,"L{F3}"CHR$(227);:GOSUB 40700
     :GOTO 40100                                 <102>
5301 :                                           <197>
5302 : REM --- FOR                               <026>
5303 :                                           <199>
5315 GOSUB 40100:GOSUB 5110:IF LE%>0 THEN
     60090                                       <034>
5320 IF CH$<>"B"THEN 60020                       <242>
5325 GOSUB 4910:IF CH$<>"£"THEN 5335             <133>
5330 GOSUB 4910:GOTO 5340                        <022>
5335 PRINT#4,AR$(123);:GOSUB 40700               <190>
5340 IF CH$<>CHR$(13)AND CH$<>":"THEN 6002
     0                                           <045>
5345 GOSUB 40710:I=PR:GOSUB 40010                <003>
5355 PRINT#4,"T"CHR$(L%)"{SHIFT-SPACE}"CHR
     $(H%)AR$(124);:RETURN                       <215>
5401 :                                           <043>
5402 : REM --- NEXT                              <215>
5403 :                                           <045>
```

```
5415 PRINT#4,AR$(125);:GOSUB 40700                <032>
5420 GET#2,CH$:IF CH$=":"OR CH$=CHR$(13)TH
     EN RETURN                                    <017>
5425 IF CH$>"{CTRL-F}"THEN 60020                  <110>
5430 I%=3-(CH$>"{CTRL-C}")                        <203>
5435 GET#2,CH$:I%=I%-1:IF I%THEN 5435             <095>
5440 IF CH$=","THEN 5415                          <228>
5445 IF CH$=":"OR CH$=CHR$(13)THEN RETURN         <087>
5450 GOTO 60020                                   <248>
5501 :                                            <143>
5502 : REM --- INPUT#                             <137>
5503 :                                            <145>
5515 GOSUB 4810:IF CH$<>","THEN 60020             <223>
5520 PRINT#4,AR$(130);:GOSUB 40700:GOSUB 4
     0100:GOSUB 5640:PRINT#4," JG";:GOTO 4
     0700                                         <035>
5601 :                                            <243>
5602 : REM --- INPUT                              <187>
5603 :                                            <245>
5615 GOSUB 40100:IF CH$<>"{CTRL-B}"THEN 56
     40                                           <148>
5620 GOSUB 40100:I$=CH$:GOSUB 40100:J$=CH$
     :GOSUB 40100                                 <065>
5625 IF CH$<>";"THEN LE%=2:PL%=ASC(I$):PH%
     =ASC(J$):GOTO 5645                           <181>
5630 CH$=J$:PF%=-1:GOSUB 5025:PRINT#4,AR$(
     126);:GOSUB 40700                            <178>
5640 GOSUB 5010                                   <072>
5645 PRINT#4," "CHR$(249)"G";:IF PH%<>-1 T
     HEN 5655                                     <152>
5650 PRINT#4,CHR$(104)CHR$(168)CHR$(104);:
     PR=PR+6.:GOTO 5660                           <043>
5655 PRINT#4,CHR$(169)CHR$(PL%)CHR$(160)CH
     R$(PH%);:GOSUB 40710                         <047>
5660 PRINT#4,AR$(LE%+127);:GOSUB 40700:IF
     CH$=","THEN GOSUB 40100:GOTO 5640            <049>
5665 RETURN                                       <135>
5701 :                                            <089>
5702 : REM --- READ                               <153>
5703 :                                            <091>
5715 GOSUB 40100:IF CH$>"{CTRL-F}"THEN 600
     20                                           <101>
5720 PF%=-1:GOSUB 5012:PRINT#4,AR$(LE%+131
     );:GOSUB 40700                               <142>
5725 IF LE%=3 THEN 60020                          <211>
5730 IF CH$=","THEN 5715                          <056>
5735 RETURN                                       <205>
5801 :                                            <189>
5802 : REM --- LET                                <015>
5803 :                                            <191>
5815 GOSUB 40100:GOTO 5110                        <202>
5851 :                                            <239>
5852 : REM --- GOTO                               <005>
5853 :                                            <241>
5862 CH%=76                                       <241>
5865 GOSUB 40100:GL%=ASC(CH$):GOSUB 40100:
     GH%=ASC(CH$):ZG=GH%*256.+GL%                 <081>
5870 I%=0:IF ZG>ZN THEN 5885                      <079>
5880 GOSUB 40520:PRINT#4,CHR$(CH%)CHR$(L%)
     CHR$(H%);:GOSUB 40700:GOTO 40100             <250>
5885 PRINT#4,"↑↑"CHR$(CH%)CHR$(GL%)CH$CHR$
     (ZZ%AND 255)CHR$(ZZ%/256);                   <067>
5890 GOSUB 40700:GOTO 40100                       <215>
5901 :                                            <035>
5902 : REM --- IF                                 <076>
5903 :                                            <037>
5915 GOSUB 4810:PRINT#4,AR$(101)"P{CTRL-C}
     ";:GOSUB 40300:PR=PR+8.                      <183>
5925 IF CH$<>CHR$(167)THEN 60020                  <104>
5930 CH$=":":RETURN                               <180>
5951 :                                            <085>
5952 : REM --- RESTORE                            <064>
5953 :                                            <087>
5965 PRINT#4," ;{CTRL-H}";:GOSUB 40700:GOT
     O 40100                                      <040>
6001 :                                            <135>
6002 : REM --- GOSUB                              <197>
6003 :                                            <137>
6015 CH%=32:GOTO 5865                             <116>
6051 :                                            <185>
6052 : REM --- RETURN                             <217>
6053 :                                            <187>
6065 PRINT#4,CHR$(96);:PR=PR+1.                   <134>
6070 GOTO 40100                                   <248>
6101 :                                            <235>
6102 : REM --- STOP                               <006>
6103 :                                            <237>
6120 PRINT#4,"8LK£";:GOSUB 40720:GOTO 4010
     0                                            <071>
6151 :                                            <031>
6152 : REM --- ON X GOTO/GOSUB                    <146>
6153 :                                            <033>
6162 GOSUB 4810:CH%=ASC(CH$):IF CH%=137 TH
     EN CH%=76:GOTO 6170                          <185>
6164 IF CH%=141 THEN CH%=32:GOTO 6170             <053>
6166 GOTO 60020                                   <202>
6170 PRINT#4,AR$(88)AR$(84)"{F1}"CHR$(98)"
     {CTRL-J}"CHR$(101)CHR$(98);                  <126>
6180 IF CH%=32 THEN PRINT#4,"{CTRL-J}";:PR
     =PR+1.                                       <083>
6185 I=PR+15.:GOSUB 40010:PRINT#4,CHR$(141
     )HL$"{BLACK}"                                <047>
6190 GOSUB 40250:PR=PR+19.                        <193>
6192 IF CH%=32 THEN PRINT#4,CHR$(234)CHR$(
     234)CHR$(234);:GOSUB 40700                   <085>
6195 GOSUB 5865:IF CH%=32 THEN GOSUB 40250
     :GOSUB 40700                                 <039>
6200 IF CH$=","THEN 6195                          <187>
6210 IF CH$=":"THEN GOSUB 40900:OG%=OG%+1         <045>
6220 RETURN                                       <182>
6251 :                                            <131>
6252 : REM --- WAIT                               <145>
6253 :                                            <133>
6265 GOSUB 4810:IF CH$<>","THEN 60020             <211>
6270 GOSUB 4810:IF CH$<>","THEN 6285              <188>
6275 GOSUB 4810                                   <063>
6280 PRINT#4,AR$(97);:GOTO 40700                  <095>
6285 PRINT#4,"£"C0$AR$(65);:GOSUB 40730:GO
     TO 6280                                      <043>
6301 : REM --- LOAD                               <251>
6303 :                                            <183>
6320 GOSUB 4960                                   <028>
6325 IF CH$=","THEN GOSUB 4810:GOTO 6340          <033>
6330 PRINT#4,"£{CTRL-A}"AR$(65);:GOSUB 407
     30:GOTO 6345                                 <014>
6340 IF CH$=","THEN GOSUB 4810:GOTO 6355          <224>
6345 PRINT#4,"£"C0$AR$(65);:GOSUB 40730           <100>
6355 PRINT#4,AR$(134);:GOTO 40700                 <204>
6401 :                                            <027>
6402 : REM --- POKE                               <124>
6403 :                                            <029>
6415 GOSUB 4510:IF S2%(1)=2 THEN 60090            <055>
6420 IF S2%(1)=0 THEN PRINT#4,AR$(135);:GO
     SUB 40700                                    <106>
6425 IF CH$<>","THEN GOTO 60020                   <178>
6430 GOSUB 4810:PRINT#4,AR$(136);:GOTO 407
     00                                           <106>
6451 :                                            <077>
6452 : REM --- PRINT#                             <165>
6453 :                                            <079>
6465 GOSUB 6665:PRINT#4," JG";:GOTO 40700         <198>
6501 :                                            <127>
6502 : REM --- PRINT                              <165>
6503 :                                            <129>
6512 CR%=-1                                       <107>
6515 GOSUB 40100                                  <201>
6517 IF CH$=";"THEN CR%=0:GOTO 6515               <255>
6520 IF CH$=","THEN CR%=0:PRINT#4,AR$(140)
     ;:GOSUB 40700:GOTO 6515                      <017>
6525 IF CH$<>CHR$(163)THEN 6540                   <008>
6530 CR%=0:GOSUB 4810:PRINT#4,AR$(138);:GO
     SUB 40700:IF CH$<>")"THEN 60020              <132>
6535 GOTO 6515                                    <086>
6540 IF CH$<>CHR$(166)THEN 6552                   <081>
6545 CR%=0:GOSUB 4810:PRINT#4,AR$(139);:GO
     SUB 40700:IF CH$<>")"THEN GOTO 60020         <105>
6550 GOTO 6515                                    <101>
6552 IF CH$=CHR$(13)OR CH$=":"THEN 6575           <043>
6555 CR%=-1:FL%=0:FU%=-1:RE%=0:S1%=0:S2%=0
     :KO%=0:KL%=0:SR%=-1:GOSUB 4520               <215>
```

Listing 3. (Fortsetzung)

```
6560 IF S2%(1)=1 THEN GOSUB 4920                    <135>
6565 IF S2%(1)<2 THEN PRINT#4,AR$(137);:GO
     SUB 40700:GOTO 6517                            <039>
6570 PRINT#4,AR$(141);:GOSUB 40700:GOTO 65
     17                                             <202>
6575 IF NOT CR%THEN RETURN                          <124>
6580 PRINT#4,AR$(80);:GOTO 40700                    <111>
6601 :                                              <227>
6602 : REM --- CLR                                  <179>
6603 :                                              <229>
6615 PRINT#4," "CHR$(231)"↑ "CHR$(20)"{CTR
     L-H}";:PR=PR+6.                                <191>
6620 GOTO 40100                                     <034>
6651 :                                              <021>
6652 : REM --- CMD                                  <202>
6653 :                                              <023>
6665 GOSUB 4810:PRINT#4,AR$(142);:GOSUB 40
     700:IF CH$<>","THEN 6580                       <185>
6670 GOTO 6512                                      <030>
6701 :                                              <073>
6702 : REM --- SYS                                  <048>
6703 :                                              <075>
6715 GOSUB 4510:IF S2%(1)=2 THEN 60090              <101>
6720 IF S2%(1)=0 THEN PRINT#4,AR$(135);:GO
     SUB 40700                                      <152>
6725 PRINT#4,AR$(94);:GOTO 40700                    <020>
6751 :                                              <123>
6752 : REM --- OPEN                                 <116>
6753 :                                              <125>
6765 GOSUB 4810:IF CH$=","THEN 6790                 <227>
6770 PRINT#4,"£{CTRL-A}"AR$(65);:GOSUB 407
     30                                             <035>
6775 PRINT#4,"£"C0$AR$(65);:GOSUB 40730             <022>
6780 PRINT#4,"£2{SHIFT-SPACE}█"AR$(110);:G
     OSUB 40710                                     <024>
6785 PRINT#4,AR$(78);:GOTO 40700                    <092>
6790 GOSUB 4810:IF CH$<>","THEN 6775                <224>
6795 GOSUB 4810:IF CH$<>","THEN 6780                <085>
6800 GOSUB 4960:GOTO 6785                           <002>
6851 :                                              <223>
6852 : REM --- CLOSE                                <032>
6853 :                                              <225>
6865 GOSUB 4810:PRINT#4,AR$(76);:GOTO 4070
     0                                              <080>
6901 :                                              <017>
6902 : REM --- GET                                  <224>
6903 :                                              <019>
6915 GOSUB 40100:IF CH$>"{CTRL-F}"THEN 600
     20                                             <031>
6920 PF%=-1:GOSUB 5012:PRINT#4,AR$(70+2*LE
     %);:GOSUB 40700                                <051>
6925 IF LE%=3 THEN 60020                            <141>
6930 IF CH$=","THEN 6915                            <026>
6935 RETURN                                         <135>
6951 :                                              <069>
6952 : REM --- GET#                                 <160>
6953 :                                              <071>
6960 GOSUB 4810:IF CH$<>","THEN 60020               <144>
6965 PRINT#4,AR$(130);:GOSUB 40700                  <022>
6970 GOSUB 6915:PRINT#4," ‾J‾B";:GOTO 40700         <083>
7500 :                                              <110>
8000 :                                              <102>
8001 REM ***** PASS 2 HAUPTPROGRAMM                 <053>
8002 :                                              <104>
8010 GOSUB 3010:ZN=0:OG%=0:ZH$=C0$:ZL$=C0$
     :ZZ%=0                                         <000>
8015 GOSUB 4710:IF FL%THEN 8050                     <238>
8020 GOSUB 40100:IF CH$>CHR$(127)AND CH$<C
     HR$(163)THEN GOSUB 5210:GOTO 8030              <202>
8025 IF CH$<"{CTRL-H}"THEN GOSUB 5110               <023>
8030 IF CH$=":"THEN 8020                            <219>
8035 IF CH$=CHR$(13)THEN 8015                       <121>
8040 GOTO 60020                                     <042>
8050 PRINT#4,"L{F3}"CHR$(227);:GOSUB 40700          <012>
8055 PRINT"{DOWN}PROGRAMM{2SPACE}:"PR-P2"
     BYTES"                                         <127>
8057 PRINT"ZEILEN{4SPACE}:"ZZ%/4-OG%                <048>
8060 IF PR+256>VA THEN GOTO 60070                   <180>
8065 CLOSE 4:CLOSE 2:OPEN 1,8,15,"S:£D,£K,
     £1":CLOSE 1                                    <210>
8070 :                                              <172>
```

```
8071 REM ***** PASS 3                               <198>
8072 :                                              <174>
8100 PRINT"{DOWN,YELLOW}PASS 3{LIG.BLUE}":
     OPEN 2,8,2,"£2,P,R":OPEN 4,8,4,"£3,P,
     W":NZ%=0:JA=P2                                 <109>
8102 IF NZ%=ZZ%THEN NA=60000:GOTO 8110              <061>
8105 NZ=PEEK(49152+NZ%)+PEEK(49153+NZ%)*25
     6.:PRINT NZ"{UP}"                              <164>
8106 NA=PEEK(49154+NZ%)+PEEK(49155+NZ%)*25
     6.:NZ%=NZ%+4                                   <016>
8107 IF JA>NA THEN 8102                             <101>
8110 JA=JA+1:GOSUB 40100:I%=ST:IF CH$<"↑"T
     HEN 8145                                       <151>
8115 JA=JA+1:GOSUB 40100:IF CH$<"↑"THEN PR
     INT#4,"↑";:GOTO 8145                           <235>
8120 GOSUB 40100:I$=CH$:GOSUB 40100:GL%=AS
     C(CH$):GOSUB 40100:GH%=ASC(CH$):JA=JA
     +1                                             <007>
8125 GOSUB 40510:PRINT#4,I$CHR$(L%)CHR$(H%
     );:GOTO 8107                                   <244>
8145 PRINT#4,CH$;:IF I%<>64 THEN 8107               <171>
8147 :                                              <249>
8150 CLOSE 4:CLOSE 2:OPEN 1,8,15,"S:C-"+LE
     FT$(NA$,14)+",£2"                              <156>
8510 PRINT#1,"C:C-"+LEFT$(NA$,14)+"=£S,RUN
     TIME,£X,£3"                                    <234>
8520 PRINT#1,"S:£S,£X,£3":CLOSE 1:RETURN            <207>
40001 :                                             <098>
40002 : REM ***** HILFSROUTINEN                     <096>
40003 :                                             <100>
40010 H%=I/256:L%=I-H%*256.:HL$=CHR$(L%)+C
      HR$(H%):RETURN                                <027>
40100 GET#2,CH$:IF CH$=""THEN CH$=C0$               <033>
40110 RETURN                                        <033>
40200 GOSUB 40010                                   <135>
40210 PRINT#3,HL$;:RETURN                           <033>
40250 I=ZN:GOTO 40310                               <219>
40300 I=ZN+1.                                       <063>
40310 GOSUB 40010:PRINT#4,"↑↑L"HL$CHR$(ZZ%
      AND 255)CHR$(ZZ%/256);:RETURN                 <115>
40510 GOSUB 40100:L%=ASC(CH$):GOSUB 40100:
      I%=L%+256*ASC(CH$)                            <110>
40520 IF I%>=ZZ%THEN I=PR-3.:GOTO 40010             <104>
40530 J%=PEEK(49153+I%):IF J%<GH%THEN I%=I
      %+4:GOTO 40520                                <075>
40540 IF J%=GH%THEN IF PEEK(49152+I%)<GL%T
      HEN I%=I%+4:GOTO 40520                        <030>
40550 L%=PEEK(49154+I%):H%=PEEK(49155+I%):
      RETURN                                        <034>
40560 :                                             <149>
40600 GOSUB 40100:ZL$=CH$:GOSUB 40100:ZH$=
      CH$:RETURN                                    <216>
40630 :                                             <219>
40700 PR=PR+3.:RETURN                               <021>
40710 PR=PR+7.:RETURN                               <037>
40720 PR=PR+4.:RETURN                               <044>
40730 PR=PR+5.:RETURN                               <055>
40800 S1%=S1%+1:S1%(S1%,0)=1:S1%(S1%,1)=15
      :S1%(S1%,2)=1:S1%(S1%,3)=1:KO%=KO%+1          <160>
40810 RETURN                                        <227>
40890 :                                             <225>
40891 REM ===== ZEILENNUMMER SPEICHERN              <109>
40892 :                                             <227>
40900 I=PR:GOSUB 40010:POKE 49154+ZZ%,L%:P
      OKE 49155+ZZ%,H%:L%=ASC(ZL$)                  <185>
40910 H%=ASC(ZH$):POKE 49152+ZZ%,L%:POKE 4
      9153+ZZ%,H%:ZZ%=ZZ%+4:RETURN                  <009>
60000 :                                             <029>
60001 REM ***** FEHLERMELDUNGEN                     <123>
60002 :                                             <031>
60020 PRINT"{YELLOW,DOWN}SYNTAX FEHLER !!{
      LIG.BLUE}":GOTO 60080                         <027>
60030 PRINT"{YELLOW,DOWN}FEHLERHAFTE DATAZ
      EILE !!{LIG.BLUE}":GOTO 60080                 <122>
60040 PRINT"{YELLOW,DOWN}'(' WIRD NACH FUN
      KTIONEN ERWARTET !!{LIG.BLUE}":GOTO
      60080                                         <220>
60050 PRINT"{YELLOW,DOWN}ILLEGALER BEFEHL
      !!{LIG.BLUE}":GOTO 60080                      <192>
60060 PRINT"{YELLOW,DOWN}ILLEGALES ZEICHEN
      !!{LIG.BLUE}":GOTO 60080                      <195>
60070 PRINT"{YELLOW,DOWN}SPEICHER IST VOLL
```

```
      !!{LIG.BLUE}"                       <246>
60080 CLOSE 2:CLOSE 3:CLOSE 4:CLOSE 5:END <231>
60090 PRINT"{YELLOW,DOWN}FALSCHER VARIABLE
      N-/KONSTANTENTYP !!{LIG.BLUE}":GOTO
      60080                               <147>
60100 PRINT"{YELLOW,DOWN}KLAMMER-/KOMMAFEH
      LER !!{LIG.BLUE}":GOTO 60080        <194>
60110 PRINT"{YELLOW,DOWN}UNGUELTIGE SPRUNG
      ANWEISUNG !!{LIG.BLUE}":GOTO 60080  <251>
60120 PRINT"{YELLOW,DOWN}TI/ST KANN NICHT
      BESCHRIEBEN WERDEN !!{LIG.BLUE}":GOT
      O 60080                             <116>
60130 PRINT"{YELLOW,DOWN}FALSCHES GOTO/GOS
      UB !!{LIG.BLUE}":GOTO 60080         <084>
61001 :                                   <014>
61010 DATA 162,251,160,159,76,212,187     <013>
61017 :                                   <030>
61018 REM  ADRESSEN DER MASCHINEN-        <201>
61019 REM  ROUTINEN IN 'RUNTIME'          <125>
61020 :                                   <033>
61100 DATA  25, 9                         <078>
61110 DATA 195,11,192,11,186,11,169,10    <012>
61120 DATA 213,11,210,11,204,11,193,10    <194>
61130 DATA 231,11,228,11,222,11,220,10    <019>
61140 DATA 249,11,246,11,240,11, 42,11    <042>
61150 DATA  11,12,  8,12,  2,12, 50,13    <201>
61160 DATA  29,12, 26,12, 20,12,107,10    <225>
61170 DATA  47,12, 44,12, 38,12,130,10    <156>
61180 DATA 204,12,201,12,195,12, 65,10    <245>
61190 DATA 220,12,217,12,211,12, 72,10    <198>
61200 DATA 236,12,233,12,227,12, 79,10    <178>
61210 DATA 252,12,249,12,243,12, 86,10    <167>
61220 DATA  12,13,  9,13,  3,13, 93,10    <196>
61230 DATA  28,13, 25,13, 19,13,100,10    <194>
61240 DATA  65,12,121,11                  <038>
61250 DATA  56,12,153,10                  <020>
61260 DATA  59,13, 82,13 :REM SGN         <140>
61270 DATA 123,13,  0, 0 :REM INT         <107>
61280 DATA 132,13,141,13 :REM ABS         <004>
61290 DATA 167,13,  0, 0 :REM USR         <070>
61300 DATA 158,19,176,13 :REM BYTE,FRE    <229>
61310 DATA 118,19,181,13 :REM LET 2,POS   <171>
61320 DATA 192,13,106,19 :REM SQR,GET F   <071>
61330 DATA 255,13, 81,19 :REM RND,GET I   <239>
61340 DATA   8,14, 66,19 :REM LOG,GET S   <140>
61350 DATA  17,14, 50,19 :REM EXP,CLOSE   <213>
61360 DATA  26,14, 33,19 :REM COS,OPEN    <207>
61370 DATA  35,14, 28,19 :REM SIN,<CR>    <059>
61380 DATA  44,14,131, 8 :REM TAN,PUSHA   <217>
61390 DATA  53,14,144, 8 :REM ATN,POPA    <196>
61400 DATA  39,16, 74,14 :REM PEEK        <223>
61410 DATA 165,14,148, 8 :REM LEN,INCST   <240>
61420 DATA 245,14,105,14 :REM STR$,PUTS   <115>
61430 DATA  42,15, 83,14 :REM VAL,GLEN    <168>
61440 DATA 210,14, 15,19 :REM ASC,SYS     <043>
61450 DATA  74,12,182,14 :REM TOSS,CHR$   <070>
61460 DATA 251,17, 69, 9 :REM WAI,LEFT$   <200>
61470 DATA  68, 8,110, 9 :REM (),RIGHT$   <039>
61480 DATA 238,17,197, 9 :REM IF,MID$     <191>
61490 DATA 138,11,170,11 :REM FLP<=>INT   <253>
61500 DATA 192, 8,216, 8,238, 8:REM X()   <161>
61510 DATA 197, 8,221, 8,243, 8:REM X     <225>
61520 DATA  79,12, 90,12, 96,12:REM ST    <169>
61530 DATA 253,15,  4,16, 11,16:REM > <   <175>
61540 DATA  18,16, 25,16, 32,16:REM >=    <077>
61550 DATA  76,16, 57,16, 92,16:REM LET   <084>
61560 DATA 119,16,126,16,151,16:REM FOR   <102>
61570 DATA 240,16              :REM "X"   <176>
61580 DATA  34,17, 58,17, 11,17:REM INP   <153>
61590 DATA  76,17              :REM IN#   <047>
61600 DATA  88,17,162,17,212,17:REM REA   <237>
61610 DATA  26,18,122,18,138,18:REM LOA   <163>
61620 DATA 160,18,185,18,206,18:REM PRI   <099>
61630 DATA 215,18,229,18       :REM PRI   <195>
61640 DATA   3, 19             :REM PR#   <052>
61697 :                                   <204>
61700 DATA 3,3,3,3,3,3,0,0,1,1,1,1        <094>
61710 DATA 0,0,0,0,0,0,0,0,0,0,0,0        <071>
61720 DATA 0,0,0,0,3,1,2,1,0,2,2,0        <205>
61730 DATA 2,1,1,2,1,2,1,2,1,2            <116>
61800 DATA 7,7,8,8,10,4,3                 <148>
61900 DATA 1,8,12,8,194,7,158,32,50,48    <196>
61910 DATA 54,50,0,0,0,32,20,8,76,0,0     <186>
61920 DATA 169,0,141,31,8,162,0,169,0     <044>
61930 DATA 157,0,0,224,0,208,7,172,31,8   <063>
61940 DATA 192,160,240,8,232,208,239      <115>
61950 DATA 238,31,8,208,234,169,0,133     <147>
61960 DATA 51,169,0,133,52,169,0,133,65   <247>
61970 DATA 169,0,133,66,96                <254>
```

Listing 3. (Schluß)

```
Name : runtime                 3185 3ce5
----------------------------------------
3185 : 84 32 a0 00 b1 31 85 47   94
318d : c8 b1 31 85 48 c8 b1 31   1f
3195 : a8 84 20 b1 31 85 28 c8   2f
319d : b1 31 85 29 20 90 08 85   1f
31a5 : 72 20 90 08 85 71 20 57   60
31ad : b3 18 8a 65 47 85 47 98   ab
31b5 : 65 48 85 48 a4 20 88 88   27
31bd : c0 01 d0 d5 60 a6 33 d0   96
31c5 : 02 c6 34 c6 33 a2 00 81   5c
31cd : 33 60 a2 00 a1 33 e6 33   8f
31d5 : d0 02 e6 34 60 a9 fb 20   6a
31dd : a7 08 a6 33 a4 34 4c d4   5f
31e5 : bb 18 65 33 85 33 b0 02   25
31ed : c6 34 60 20 74 0d a9 05   4a
31f5 : 18 65 33 85 33 90 02 e6   cb
31fd : 34 60 20 44 08 f0 04 85   15
3205 : 47 84 48 a9 fb 20 a7 08   45
320d : a0 04 b1 47 91 33 88 10   fa
3215 : f9 60 20 44 08 f0 04 85   f2
321d : 47 84 48 a0 00 b1 47 20   b7
3225 : 83 08 c8 b1 47 4c 83 08   0a
322d : 20 44 08 f0 04 85 47 84   22
3235 : 48 a9 fc 20 a7 08 a0 03   d8
323d : a2 00 a1 47 91 33 88 e6   d3
3245 : 47 d0 02 e6 48 a5 47 91   44
324d : 33 88 a5 48 91 33 a9 02   94
3255 : 81 33 60 a0 00 b1 33 a8   48
325d : c8 98 c8 18 71 33 48 88   89
3265 : b1 33 c8 91 33 88 88 d0   4f
326d : f7 68 c8 91 33 4c 94 08   f5
3275 : a0 01 b1 33 85 20 c8 b1   48
327d : 33 a8 a9 03 4c b6 08 20   aa
3285 : 36 09 a2 02 a5 20 38 f1   49
328d : 33 f0 06 b0 0e a5 20 91   82
3295 : 33 88 88 98 20 b6 08 8a   2e
329d : 4c 83 08 85 20 88 88 88   d7
32a5 : 30 f5 e8 e8 e8 4c 4a 09   53
32ad : 20 36 09 84 21 a0 02 b1   3e
32b5 : 33 38 e5 20 90 34 18 88   9e
32bd : 71 33 91 33 88 90 06 b1   1b
32c5 : 33 69 00 91 33 c8 c8 a5   c7
32cd : 20 91 33 98 aa c4 21 f0   cd
32d5 : 16 b1 33 84 20 a4 21 91   f0
32dd : 33 c6 21 a4 20 88 10 f1   bb
32e5 : 38 a5 21 20 b7 08 8a 4c   bb
32ed : 83 08 a5 20 38 f1 33 85   cd
32f5 : 20 c4 21 d0 04 98 4c 83   17
32fd : 08 c8 c8 c8 4c 75 09 a0   8a
3305 : 01 b1 33 48 38 e9 01 a0   cd
330d : 03 18 71 33 85 20 c8 b1   bf
3315 : 33 a8 a9 05 20 b6 08 20   bf
331d : 48 09 68 85 20 20 90 08   0a
3325 : 85 21 4c 73 09 86 20 84   0b
332d : 21 a0 00 b1 33 a0 02 51   b7
3335 : 33 10 23 a2 00 b1 33 10   08
333d : 02 a2 01 8a a0 ff 29 03   d7
3345 : c5 20 f0 05 c5 21 f0 01   22
334d : c8 98 a0 02 91 33 c8 91   c3
3355 : 33 a9 02 4c b6 08 a0 00   95
335d : b1 33 a0 02 d1 33 d0 08   1a
3365 : a0 01 b1 33 a0 03 d1 33   28
336d : 08 f0 0c a0 00 b1 33 10   7f
3375 : 06 68 49 01 4c 02 0a 68   f0
337d : 4c 02 0a a2 00 a0 00 4c   3f
3385 : eb 09 a2 01 a0 01 4c eb   d9
338d : 09 a2 03 a0 02 4c eb 09   00
3395 : a2 03 a0 00 4c eb 09 a2   6f
339d : 03 a0 01 4c eb 09 a2 01   4e
33a5 : a0 00 4c eb 09 a0 00 b1   cf
33ad : 33 a0 02 31 33 91 33 a0   a5
33b5 : 01 b1 33 a0 03 31 33 91   19
33bd : 33 4c 17 0a a0 00 b1 33   55
33c5 : a0 02 11 33 91 33 a0 01   48
33cd : b1 33 a0 03 11 33 91 33   f8
33d5 : 4c 17 0a a0 00 b1 33 49   30
33dd : ff 91 33 c8 b1 33 49 ff   65
33e5 : 91 33 60 18 a0 01 b1 33   6a
33ed : a0 03 71 33 91 33 a0 00   07
33f5 : b1 33 a0 02 71 33 91 33   06
33fd : 4c 17 0a 38 a0 03 b1 33   ae
3405 : a0 01 f1 33 a0 03 91 33   d7
340d : 88 b1 33 a0 00 f1 33 a0   ec
3415 : 02 91 33 4c 17 0a a0 00   7b
341d : b1 33 a0 02 51 33 08 a2   e4
3425 : 00 a0 01 20 11 0b a2 49   40
342d : a0 03 20 11 0b 20 57 b3   f0
3435 : 28 10 0c 8a 49 ff aa 98   2a
343d : 49 ff a8 e8 d0 01 c8 98   36
3445 : a0 02 91 33 c8 8a 91 33   3f
344d : 4c 17 0a b1 33 95 28 88   6f
3455 : b1 33 95 29 10 0d 38 a9   c8
```

**Listing 4. »Runtime«.
Die Maschinensprachroutinen von Hypra-Comp.
Bitte mit dem MSE (Seite 159) eingeben.**

```
345d : 00 f5 28 95 28 a9 00 f5  d0
3465 : 29 95 29 60 a0 00 b1 33  e6
346d : a0 02 51 33 08 a2 00 a0  a0
3475 : 01 20 11 0b a2 49 a0 03  29
347d : 20 11 0b a9 00 85 20 85  d6
3485 : 21 a0 10 06 71 26 72 b0  2f
348d : 0a 88 d0 f7 4c 72 0b 06  9f
3495 : 71 26 72 26 20 26 21 38  a3
349d : a5 20 e5 28 aa a5 21 e5  f9
34a5 : 29 90 06 86 20 85 21 e6  e9
34ad : 71 88 d0 e3 a4 72 a6 71  6e
34b5 : 4c f6 0a a0 01 38 a9 00  8b
34bd : f1 33 91 33 88 a9 00 f1  cc
34c5 : 33 91 33 60 20 b1 08 20  8a
34cd : 9b bc a5 65 20 83 08 a5  66
34d5 : 64 4c 83 08 a0 00 b1 33  78
34dd : aa c8 b1 33 a8 8a 20 91  41
34e5 : b3 4c 17 0a 20 9a 0b 4c  61
34ed : 9b 08 20 b1 08 a5 33 a4  8e
34f5 : 34 4c 8c ba 20 23 0d 4c  b2
34fd : c3 0b 20 aa 0b 20 b0 0b  2e
3505 : 20 6a b8 4c a0 08 20 23  23
350d : 0d 4c d5 0b 20 aa 0b 20  db
3515 : b0 0b 20 53 b8 4c a0 08  3e
351d : 20 23 0d 4c e7 0b 20 aa  48
3525 : 0b 20 b0 0b 20 2b ba 4c  ad
352d : a0 08 20 23 0d 4c f9 0b  6f
3535 : 20 aa 0b 20 b0 0b 20 12  79
353d : bb 4c a0 08 20 23 0d 4c  2f
3545 : 0b 0c 20 aa 0b 20 b0 0b  3e
354d : 20 7b bf 4c a0 08 20 23  b5
3555 : 0d 4c 1d 0c 20 aa 0b 20  15
355d : b0 0b 20 e9 af 4c a0 08  c8
3565 : 20 23 0d 4c 2f 0c 20 aa  0d
356d : 0b 20 b0 0b 20 e6 af 4c  a6
3575 : a0 08 20 74 0d 20 d4 ae  32
357d : 4c a0 08 a0 01 b1 33 49  2c
3585 : 80 91 33 60 a9 00 85 46  e4
358d : 60 20 b7 ff 20 83 08 a9  7d
3595 : 00 4c 83 08 20 7b af 4c  d2
359d : 9b 08 20 84 af 84 5e 88  7f
35a5 : 84 71 a0 06 84 5d a0 24  c9
35ad : 20 68 be a5 46 38 e9 06  3f
35b5 : 85 46 aa a0 00 b9 ff 00  ea
35bd : 9d 00 cf e8 c8 c0 06 d0  b7
35c5 : f4 a9 fc 20 a7 08 a9 06  3e
35cd : a0 03 91 33 88 a5 46 91  ac
35d5 : 33 88 a9 cf 91 33 88 a9  d9
35dd : 02 91 33 60 86 20 84 21  3e
35e5 : 20 b1 08 a5 33 a4 34 20  fe
35ed : 5b bc 08 68 29 03 c9 02  8b
35f5 : 90 02 29 02 49 01 48 20  0f
35fd : 90 08 4c 3d 0a 20 23 0d  94
3605 : 4c cc 0c 20 aa 0b a2 00  4c
360d : a0 00 4c a2 0c 20 23 0d  7d
3615 : 4c dc 0c 20 aa 0b a2 01  66
361d : a0 01 4c a2 0c 20 23 0d  0e
3625 : 4c ec 0c 20 aa 0b a2 03  82
362d : a0 02 4c a2 0c 20 23 0d  9e
3635 : 4c fc 0c 20 aa 0b a2 03  9a
363d : a0 00 4c a2 0c 20 23 0d  ad
3645 : 4c 0c 0d 20 aa 0b a2 03  72
364d : a0 01 4c a2 0c 20 23 0d  3e
3655 : 4c 1c 0d 20 aa 0b a2 01  86
365d : a0 00 4c a2 0c 20 b1 08  fd
3665 : 20 0c bc 20 aa 0b 20 fc  3c
366d : bb 4c 9b 08 20 9a 0b 20  79
3675 : 26 0d 4c 0b 0c a0 00 b1  c0
367d : 33 f0 10 c8 b1 33 29 80  a0
3685 : 91 33 a9 00 c8 91 33 c0  82
368d : 04 d0 f9 60 a0 00 b1 33  bb
3695 : 10 08 a9 ff 91 33 c8 91  0d
369d : 33 60 d0 0a c8 b1 33 d0  fe
36a5 : 04 88 98 f0 ef 88 98 91  fa
36ad : 33 c8 98 91 33 60 a5 33  d0
36b5 : a4 34 4c a2 bb 20 74 0d  83
36bd : 20 cc bc 4c a0 08 a0 01  cb
36c5 : b1 33 29 7f 91 33 60 a0  c0
36cd : 00 b1 33 10 13 c8 b1 33  19
36d5 : 49 ff 18 69 01 91 33 88  cc
36dd : b1 33 49 ff 69 00 91 33  bd
36e5 : 60 20 74 0d 20 10 03 4c  3b
36ed : a0 08 a0 00 4c 68 0d a0  37
36f5 : 01 a5 d3 91 33 88 98 91  ed
36fd : 33 60 20 74 0d 20 2b bc  ef
3705 : f0 34 10 03 4c 48 b2 20  86
370d : a0 08 a5 61 38 e9 81 08  30
3715 : 4a 18 69 01 28 90 02 69  c8
371d : 7f 85 61 a0 04 84 20 20  f0
3725 : c7 bb a5 33 a4 34 20 0f  24
372d : bb a9 5c a0 00 20 67 b8  f8
3735 : c6 61 c6 20 d0 e9 4c a0  30
373d : 08 20 74 0d 20 97 e0 4c  ef
3745 : a0 08 20 74 0d 20 ea b9  71
374d : 4c a0 08 20 74 0d 20 ed  fc
3755 : bf 4c a0 08 20 74 0d 20  7e
375d : 64 e2 4c a0 08 20 74 0d  c7
3765 : 20 6b e2 4c a0 08 20 74  31
376d : 0d 20 b4 e2 4c a0 08 20  3e
3775 : 74 0d 20 0e e3 4c a0 08  6d
377d : a0 01 b1 33 85 31 88 b1  d8
3785 : 33 85 32 60 20 3e 0e b1  a3
378d : 31 c8 4c b9 0d a0 00 b1  a6
3795 : 33 a8 c8 a9 00 18 71 33  71
379d : 90 03 4c 58 b6 88 88 88  b0
37a5 : d0 f4 60 85 31 84 32 a0  f9
37ad : 00 b1 33 a8 88 b1 33 85  56
37b5 : 21 c8 b1 33 85 20 c8 b1  ed
37bd : 33 84 45 a8 f0 08 88 b1  6e
37c5 : 20 91 31 98 d0 f8 a4 45  ff
37cd : c0 03 f0 12 a5 31 18 71  b4
37d5 : 33 85 31 90 02 e6 32 88  5a
37dd : 88 88 88 4c 72 0e 60 20  ae
37e5 : 53 0e aa b1 33 20 b6 08  3f
37ed : 98 91 33 c8 8a 91 33 60  f6
37f5 : a0 01 b1 33 c6 46 a6 46  ae
37fd : 9d 00 cf 98 91 33 88 8a  8b
3805 : 91 33 a9 cf 20 83 08 a9  26
380d : 02 4c 83 08 a0 00 b1 33  4e
3815 : a8 c8 b1 33 d0 08 88 88  75
381d : 88 d0 f7 4c 48 b2 88 b1  34
3825 : 33 85 31 88 b1 33 85 32  a7
382d : a0 00 b1 31 4c a8 0e 20  e2
3835 : 74 0d a0 00 20 df bd a5  9b
383d : 46 38 e9 0b 85 46 a0 00  88
3845 : aa b9 ff 00 f0 07 9d 00  89
384d : cf c8 e8 d0 f4 98 a0 04  73
3855 : 91 33 88 a5 46 91 33 88  25
385d : a9 cf 91 33 88 a9 02 91  ba
3865 : 33 4c 94 08 20 53 0e 85  c4
386d : 20 a8 d0 06 20 f7 b8 4c  13
3875 : 56 0f a9 00 20 83 08 a5  47
387d : 33 38 e5 20 85 7a a5 34  75
3885 : e9 00 85 7b a8 a5 7a 20  21
388d : 94 08 20 69 0e 20 34 11  2f
3895 : 20 5c 0f 4c 9b 08 a0 00  ad
389d : b1 33 18 69 02 4c b7 08  8d
38a5 : 4c 9b 08 b5 5b d0 21 b4  a2
38ad : 59 b1 33 d0 0c 88 88 88  fe
38b5 : 98 95 59 d5 5a d0 f2 38  91
38bd : 60 95 5b 88 b1 33 95 57  89
38c5 : 88 b1 33 95 58 4c 94 0f  fe
38cd : f6 57 d0 02 f6 58 d6 5b  27
38d5 : d0 06 a9 00 b4 59 91 33  d5
38dd : 18 60 a5 45 4c b6 08 86  df
38e5 : 20 84 21 a0 00 b1 33 84  07
38ed : 5f a8 c8 84 5e c8 84 5a  56
38f5 : b1 33 38 65 5a 85 59 a8  83
38fd : 88 88 88 84 45 a9 00 85  29
3905 : 5b 85 60 a2 00 20 69 0f  54
390d : 90 13 a2 05 20 69 0f 90  1b
3915 : 04 a2 03 d0 02 a2 01 20  bf
391d : a0 0f 4c 01 0a a2 05 20  82
3925 : 69 0f 90 04 a2 00 f0 ef  88
392d : a0 00 b1 5c d1 57 f0 d3  08
3935 : 08 20 a0 0f 4c 3d 0a a2  73
393d : 00 a0 00 4c a5 0f a2 01  76
3945 : a0 01 4c a5 0f a2 03 a0  81
394d : 03 4c a5 0f a2 03 a0 00  86
3955 : 4c a5 0f a2 03 a0 01 4c  5e
395d : a5 0f a2 00 a0 01 4c a5  c1
3965 : 0f 20 b1 08 20 f7 b7 a0  d4
396d : 00 b1 14 20 83 08 a9 00  6e
3975 : 4c 83 08 85 4e 84 4f a0  bd
397d : 00 b1 33 c8 91 4e b1 33  f4
3985 : 88 91 4e 4c 17 0a 85 4e  67
398d : 84 4f a0 04 b1 33 91 4e  f9
3995 : 88 10 f9 4c b4 08 85 4e  6b
399d : 84 4f 20 53 0e a0 00 91  44
39a5 : 4e e6 4e d0 02 e6 4f a5  f4
39ad : 4e a4 4f 20 69 0e 4c 5c  16
39b5 : 0f a9 bc a0 b9 4c c5 08  01
39bd : a9 fc 20 a7 08 98 a0 00  a9
39c5 : 91 33 c8 8a 91 33 c8 a5  94
39cd : 4f 91 33 c8 a5 4e 91 33  44
39d5 : 60 a0 03 b1 33 aa 88 b1  8a
39dd : 33 a8 8a 85 49 84 4a 20  da
39e5 : a2 bb a5 33 18 69 04 a4  5b
39ed : 34 90 01 c8 20 67 b8 a2  28
39f5 : 90 a0 05 b1 33 10 02 a2  4e
39fd : b0 8e d1 10 a5 33 18 69  92
3a05 : 09 a4 34 90 01 c8 20 5b  0d
3a0d : bc f0 02 90 15 20 d0 bb  e1
3a15 : a0 00 b1 33 8d e7 10 c8  72
3a1d : b1 33 8d e6 10 68 68 4c  26
3a25 : ff ff 20 d0 bb a9 0e 4c  20
3a2d : b6 08 85 31 84 32 a0 00  cb
3a35 : b1 31 e6 31 d0 02 e6 32  7c
3a3d : aa f0 09 b1 31 20 d2 ff  37
3a45 : c8 ca d0 f7 60 85 31 84  a5
3a4d : 32 a0 00 b9 00 02 f0 06  e6
3a55 : c8 91 31 4c 11 11 98 a0  f9
3a5d : 00 91 31 60 20 28 11 4c  9e
3a65 : d0 bb 85 49 84 4a a9 00  df
3a6d : 85 7a a9 02 85 7b 20 79  82
3a75 : 00 4c f3 bc 20 28 11 20  f8
3a7d : 9b bc a0 00 a5 65 91 49  fd
3a85 : c8 a5 64 91 49 60 20 94  ac
3a8d : 08 20 90 08 aa 86 13 4c  8e
3a95 : 1e e1 20 7b 11 c9 01 90  a0
3a9d : 0e f0 03 4c 99 ad 20 8b  0c
3aa5 : 11 20 91 b3 4c d0 bb a0  1d
3aad : 05 b1 41 88 91 49 d0 f9  87
3ab5 : a9 06 4c 97 11 85 49 84  d3
3abd : 4a a0 00 b1 41 10 05 a2  7c
3ac5 : 0d 4c 37 a4 60 c8 b1 41  f0
3acd : 48 c8 b1 41 aa 68 a8 a9  f2
3ad5 : 03 18 65 41 85 41 90 02  0e
3add : e6 42 8a 60 20 7b 11 c9  49
3ae5 : 01 90 0f d0 b6 20 8b 11  c9
3aed : 98 a0 00 91 49 c8 8a 91  30
3af5 : 49 60 a5 41 a4 42 18 69  8f
3afd : 01 90 01 c8 20 a2 bb a9  f9
3b05 : 06 20 97 11 20 9b bc a5  40
3b0d : 65 a6 64 4c af 11 20 7b  63
3b15 : 11 c9 02 d0 86 a9 01 20  9f
3b1d : 97 11 b1 41 a8 b1 41 91  12
3b25 : 49 88 10 f9 38 4c 98 11  60
3b2d : 20 90 08 a8 20 90 08 aa  a8
3b35 : f0 01 60 a8 60 a0 04 b1  51
3b3d : 33 85 15 c8 b1 33 85 14  84
3b45 : a0 03 b1 33 85 49 a0 01  61
3b4d : b1 33 85 4a 20 3e b8 a9  6d
3b55 : 06 4c b6 08 20 47 12 20  f5
3b5d : 5b 12 a9 00 a6 2b a4 2c  da
3b65 : 85 7b 20 d5 ff b0 17 20  8d
3b6d : b7 ff 29 bf f0 03 4c 9c  f8
3b75 : e1 20 33 08 a5 b9 f0 03  26
3b7d : 4c 11 08 4c 0e 08 4c f9  24
3b85 : e0 a0 03 b1 33 aa a0 01  b9
3b8d : b1 33 a8 a9 00 20 00 fe  36
3b95 : a9 04 4c b6 08 20 53 0e  15
3b9d : 85 b7 a5 b7 f0 13 a5 33  03
3ba5 : 38 e5 b7 85 bb a5 34 e9  fc
3bad : 00 85 bc a8 a5 bb 20 69  3f
3bb5 : 0e 4c 5c 0f 20 b1 08 20  d2
3bbd : f7 b7 a5 14 20 83 08 a5  05
3bc5 : 15 4c 83 08 a0 03 b1 33  32
3bcd : 85 14 88 b1 33 85 15 88  79
3bd5 : b1 33 88 91 14 a9 04 4c  ab
3bdd : b6 08 20 b1 08 20 dd bd  4a
3be5 : 85 31 84 32 a0 00 b1 31  9d
3bed : f0 06 20 0c e1 c8 d0 f6  ff
3bf5 : 4c 3b ab 20 94 08 20 90  f9
3bfd : 08 38 e5 d3 90 09 aa f0  f3
3c05 : 06 20 3b ab ca d0 fa 60  3f
3c0d : 20 94 08 20 90 08 4c c4  81
3c15 : 12 a5 d3 38 e9 0a b0 fc  a1
3c1d : 49 ff 69 01 4c c4 12 a0  55
3c25 : 00 b1 33 84 5a 84 5b a8  e4
3c2d : c8 84 59 a2 00 20 69 0f  a7
3c35 : b0 08 a1 57 20 0c e1 4c  bf
3c3d : f1 12 4c 5c 0f 20 94 08  2a
3c45 : 20 90 08 aa 86 13 4c 18  67
3c4d : e1 20 90 08 85 15 20 90  06
3c55 : 08 85 14 4c 30 e1 a9 0d  81
3c5d : 4c 0c e1 20 5b 12 20 47  81
3c65 : 12 20 94 08 20 90 08 85  5f
3c6d : b8 4c c1 e1 20 94 08 20  ff
3c75 : 90 08 4c cc e1 85 49 84  2e
3c7d : 4a 4c 24 e1 20 3b 13 a0  9c
3c85 : 01 aa f0 03 91 49 98 88  4f
3c8d : 91 49 60 20 3b 13 38 e9  e0
3c95 : 30 c9 0a 90 02 a9 00 60  6d
3c9d : 20 51 13 a0 00 91 49 98  22
3ca5 : c8 91 49 60 20 51 13 a8  be
3cad : a9 00 20 91 b3 4c d0 bb  e9
3cb5 : 85 47 84 48 a9 00 a0 a0  66
3cbd : 20 5c 10 78 a9 35 85 01  7b
3cc5 : a0 00 ad 00 a0 91 47 f0  66
3ccd : 09 a8 b9 00 a0 91 47 88  5d
3cd5 : d0 f8 a9 37 85 01 58 60  f5
3cdd : 20 83 08 a9 00 4c 83 08  77
```

Listing 4. (Schluß)

17 Super-Utilities für den C 64

Wollen Sie Ihre Basic-Programme schneller und kürzer machen? Wollen Sie Betriebssystem oder Zeichensatz Ihres C64 ändern? Dieses Maschinenprogramm löst Ihre Probleme.

Das Programm Helps (Listing 1) entstand durch Zusammenfügen der in der praktischen Arbeit am häufigsten gebrauchten Maschinensprache-Routinen zu einem einzigen großen Utility-Paket.

Die beiden Bereiche Optimierung von Basic-Programmen und Zeichen-Definition stehen im Mittelpunkt. Nach ordnungsgemäßem Laden kann man ein Menü mit SYS 49152 aufrufen (Bild 1), in dem alle Unterprogramme mit Namen und SYS-Adresse genannt werden.

Nun zu den Programmen selbst:

Space-Killer (SYS 49155): Hinter diesem martialisch anmutenden Aufruf verbirgt sich nichts anderes als eine Maschinen-Routine, welche überflüssige Spaces aus einem Basic-Programm entfernt.

Nach dem SYS 49155-Aufruf erscheint in der oberen Bildschirmecke ein farbiges Symbol als Versicherung, daß alles in Ordnung ist, denn die Routine kann durchaus bis zu ¼ Stunde laufen, wenn viele Spaces zu entfernen sind und das Programm entsprechend lang ist.

Space-Expander (SYS 49158): Es erscheint auf den ersten Blick befremdend, etwas einzufügen, was man eben erst entfernt hatte. Doch die Aufgabe liegt auf der Hand: zwischen jeden Basic-Befehl ein Space einzufügen, um das Druckbild sauber und ansprechend zu gestalten.

REM-Killer (SYS 49161): erklärt sich selbst. Erwähnenswert ist die Geschwindigkeit, mit der die Routine läuft. 20 KByte-Programme werden in etwa 10 bis 30 Sekunden »gesäubert«.

Zeilen löschen (SYS 49164): Hiermit werden ganze Zeilenbereiche gelöscht. Beispiel: »SYS 49164,1000,1999« entfernt alle Basic-Zeilen mit den Nummern von 1000 bis 1999 einschließlich. Mit Druck auf die <RETURN>-Taste ist der Befehl schon ausgeführt.

Text suchen (SYS 49167,''Suchtext''): sucht eine Zeichenkette »Suchtext« im Basic-Programm und gibt die gefundenen Zeilennummern aus.
Eine Besonderheit ist das Jokerzeichen »?«: Hier wird jedes Zeichen akzeptiert, es darf allerdings nicht an erster Stelle im Suchtext stehen.

Colon-Maker (SYS 49170): ersetzt jedes Space am Anfang einer Basic-Zeile durch einen Doppelpunkt »:«.

Space-Maker (SYS 49173): ersetzt jeden Doppelpunkt am Anfang einer Basic-Zeile durch ein Space » «. Beide Routinen dienen als Strukturhilfen. Da alle der Zeilennummer unmittelbar folgenden Spaces überlesen werden, muß man erst einen »neutralen« Doppelpunkt setzen, diesen dann später durch ein Space ersetzen.

Package (SYS 49179): Hier ist die Notlösung für jeden, der einfach mehr Platz braucht und das Basic-Programm garantiert nicht mehr korrigieren muß. »Package« schmilzt nämlich benachbarte Zeilen zusammen, so daß bei jeder Verschmelzung vier Byte freiwerden (ein Null-Byte, zwei Byte der Koppeladresse plus zwei Byte der Zeilennummer, abzüglich ein Byte für den nun erforderlichen Doppelpunkt als Trennelement).

Das braucht seine Zeit, weil einige Aspekte kontrolliert werden müssen, zum Beispiel ob die zu verschmelzende Zeile angesprungen wird, dann muß sie mit ihrer Zeilennummer präsent bleiben, sonst gibt's Ärger mit dem Interpreter, der sich dann mit »UNDEF'D STATEMENT ERROR« beschwert. Für die Verschmelzungstiefe gibt es leider eine natürliche Grenze: Mehr als 255 Byte darf eine Zeile nicht lang sein, sonst kann der Interpreter die Zeilen nicht mehr binden und findet hinter einer solchen »Megazeile« keine Zeilennummer mehr.

Um Basic-Programme auf ein Minimum zu komprimieren, empfiehlt sich folgender Weg:
- zuerst SYS 49161 eingeben. Danach sind alle REMarks weg, und die folgenden Routinen haben weniger Prüfarbeit.
- dann SYS 49155 eingeben. Hier werden alle Zeilen gestaucht.

```
sys49152

sys49155: Space-Killer
sys49158: Space-Expander
sys49161: REM-Killer
sys49164: Zeilen loeschen
sys49167: Text suchen
sys49170: Colon-Maker
sys49173: Space-Maker
sys49179: Package
sys49182: CPU-Register ein
sys49185: CPU-Register aus
sys49188: BASIC-Ende setzen
sys49191: Variablen dumpen
sys49194: ROM => RAM
sys49197: Zeichensatz nach 28672
sys49200: Zeichensatz aendern
sys49203: Zeichensatz speichern
sys49206: Zeichensatz alt

ready.
█
```

Bild 1. Mit folgendem Text meldet sich »Helps« nach der Eingabe SYS 49152 <RETURN>

- zuletzt SYS 49179 eingeben, danach ist Ihr PRG redundanzfrei.

Für spätere Änderungsarbeiten wird man sich natürlich ein »augenfreundliches« Original verwahren, am besten benutzt man von allen Basic-Programmen zwei Versionen: eine originale zur Dokumentation, Korrektur, Ausdruck und eine komprimierte mit minimalen Platzansprüchen und schneller Ausführung.

CPU-Register ein (SYS 49182): zeigt fortwährend die Register des Mikroprozessors in den beiden oberen Bildschirmzeilen an.
Außerdem werden der IRQ-, NMI- und der BRK-Vektor sowie der Programmcounter angezeigt. Im Status-Register bedeutet »*« ein gesetztes Bit, ».« ein gelöschtes.

CPU-Register aus (SYS 49185): schaltet die Anzeige wieder ab.

Basic-Ende setzen (SYS 49188): setzt das Basic-Ende genau hinter das im Speicher befindliche Programm. Normalerweise erledigt das der Interpreter; wenn aber absolut geladen wurde, zum Beispiel in den $c-Block hinein, dann steht der Programm-Ende-Zeiger ebenfalls im $c-Block, und Eingabeversuche werden mit »out of memory«-Fehlermeldungen abgewehrt.

Dump (SYS 49191): gibt alle definierten Variablen unter Angabe des Namens und des aktuellen Inhalts aus.
Mit der <SHIFT>-Taste kann die Ausgabe angehalten, mit der <CBM>-Taste vorzeitig abgebrochen werden.

ROM kopieren (SYS 49194): kopiert Interpreter- und Kernel-ROM in das darunter liegende RAM. Es gibt zwar viele veröffentlichte Programme zu diesem Thema, doch ist die Geschwindigkeit dieser Programme oft nicht befriedigend.
Zeichensatz RAM: Die letzten vier Routinen befassen sich mit dem Zeichensatz:
(SYS 49197): verlegt den Zeichensatz nach $7000 und den Bildschirm nach $6c00. Die Absicht ist klar: Im RAM kann der Zeichensatz leicht verändert werden.
Zeichensatz ändern (SYS 49200): setzt voraus, daß vorher durch SYS 49197 der Zeichensatz verlegt wurde, sonst funktioniert es nicht. Es wird eine Zeichen-Änderungsroutine aktiviert, die einfach zu bedienen ist. Man fährt einfach mit dem Cursor über das zu ändernde Zeichen, drückt <RETURN>, und schon sieht man wie durch eine Lupe die Punkte-Matrix des Zeichens. Auch hier kann man den Cursor innerhalb des Zeichens frei bewegen. Will man einen Punkt löschen, drückt man die <SPACE>-Taste, zum Setzen gibt man ein Sternchen »*« ein. <RETURN> führt wieder in das Menü zurück.
Zeichensatz speichern (SYS 49203, "filename", gerätenummer): speichert den aktuellen Zeichensatz unter Angabe des Filenamens (der wie bei SAVE dem SYS-Befehl in Anführungsstrichen folgen muß), eines Kommas und der Gerätenummer entweder auf der Floppy oder dem Kassettenrecorder. Damit nicht jedesmal wieder bei Null begonnen werden muß, kann man einen Zeichensatz durch den Basic-Befehl:

```
load"filename", gerätenummer,1
```

absolut, das heißt von $7000 bis $7fff laden.
Die verbogenen Basic-Zeiger werden danach mit SYS 49188 wieder gerichtet.
Zeichensatz ROM (SYS 49206): schaltet den original Zeichensatz ab $d000 ein. Der Bildschirm wird nach $400 verlegt.

(Johannes Mockenhaupt/sk)

```
Name : helps                   c000 d000
-----------------------------------------
c000 : 4c 39 c0 4c 47 c2 4c 0e   7a
c008 : c3 4c d2 c3 4c 60 c4 4c   92
c010 : e8 c4 4c 61 c5 4c 65 c5   79
c018 : 4c 94 c5 4c c0 c7 4c 42   a9
c020 : c9 4c 30 ca 4c 58 c2 4c   a0
c028 : 39 ca 4c 16 cb 4c 6d cb   09
c030 : 4c 11 cc 4c 5a cf 4c 1a   4b
c038 : cf 20 0e c2 a0 00 a9 5e   61
c040 : a2 c0 85 fb 86 fc b1 fb   32
c048 : d0 0a a9 0e 20 16 e7 a9   ef
c050 : 08 4c 16 e7 20 d2 ff c8   2b
c058 : d0 ec e6 fc d0 e8 53 59   4c
c060 : 53 34 39 31 35 35 3a 20   68
c068 : d3 50 41 43 45 2d cb 49   9c
c070 : 4c 4c 45 52 0d 53 59 53   f5
c078 : 34 39 31 35 38 3a 20 d3   b9
c080 : 50 41 43 45 2d c5 58 50   ed
c088 : 41 4e 44 45 52 0d 53 59   38
c090 : 53 34 39 31 36 31 3a 20   88
c098 : d2 c5 cd 2d cb 49 4c 4c   37
c0a0 : 45 52 0d 53 59 53 34 39   30
c0a8 : 31 36 34 3a 20 da 45 49   c9
c0b0 : 4c 45 4e 20 4c 4f 45 53   31
c0b8 : 43 48 45 4e 0d 53 59 53   b2
c0c0 : 34 39 31 36 37 3a 20 d4   13
c0c8 : 45 58 54 20 53 55 43 48   d0
c0d0 : 45 4e 0d 53 59 53 34 39   5e
c0d8 : 31 37 30 3a 20 c3 4f 4c   ee
c0e0 : 4f 4e 2d cd 41 4b 45 52   84
c0e8 : 0d 53 59 53 34 39 31 37   a0
c0f0 : 33 3a 20 d3 50 41 43 45   69
c0f8 : 2d cd 41 4b 45 52 0d 53   87
c100 : 59 53 34 39 31 37 39 3a   5d
c108 : 20 50 41 43 4b 41 47 45   6f
c110 : 0d 53 59 53 34 39 31 38   ca
c118 : 32 3a 20 c3 d0 d5 2d d2   fe
c120 : 45 47 49 53 54 45 52 20   bf
c128 : 45 49 4e 0d 53 59 53 34   fd
c130 : 39 31 38 35 3a 20 c3 d0   0c
c138 : d5 2d d2 45 47 49 53 54   b6
c140 : 45 52 20 41 55 53 0d 53   a9
c148 : 59 53 34 39 31 38 38 3a   a9
c150 : 20 c2 c1 d3 c9 c3 2d c5   b7
c158 : 4e 44 45 20 53 45 54 5a   83
c160 : 45 4e 0d 53 59 53 34 39   ee
c168 : 31 39 31 3a 20 d6 41 52   2c
c170 : 49 41 42 4c 45 4e 20 44   44
c178 : 55 4d 50 45 4e 0d 53 59   7e
c180 : 53 34 39 31 39 34 3a 20   c0
c188 : d2 cf cd 20 3d 3e 20 d2   a5
c190 : c1 cd 0d 53 59 53 34 39   59
c198 : 31 39 37 3a 20 da 45 49   fb
c1a0 : 43 48 45 4e 53 41 54 5a   68
c1a8 : 20 4e 41 43 48 20 24 37   2d
c1b0 : 30 30 30 0d 53 59 53 34   5c
c1b8 : 39 32 30 30 3a 20 da 45   b7
c1c0 : 49 43 48 45 4e 53 41 54   93
c1c8 : 5a 20 41 45 4e 44 45 52   ec
c1d0 : 4e 0d 53 59 53 34 39 32   c5
c1d8 : 30 33 3a 20 da 45 49 43   b8
c1e0 : 48 45 4e 53 41 54 5a 20   29
c1e8 : 53 50 45 49 43 48 45 52   0e
c1f0 : 4e 0d 53 59 53 34 39 32   e5
c1f8 : 30 36 3a 20 da 45 49 43   59

c200 : 48 45 4e 53 41 54 5a 2c   61
c208 : 20 41 4c 54 0d 00 a9 0d   f8
c210 : 4c d2 ff 20 0e c2 8d 00   f6
c218 : 04 38 a5 2d e5 2b aa a5   f5
c220 : 2e e5 2c 20 cd bd a2 00   a5
c228 : bd 34 c2 d0 01 60 20 d2   03
c230 : ff e8 d0 f4 20 c2 59 54   9c
c238 : 45 53 20 d0 d2 c7 2d cc   03
c240 : 41 45 4e 47 45 0d 00 20   9d
c248 : 13 c2 a6 2b a5 2c 86 fb   99
c250 : 85 fc a0 01 b1 fb d0 29   2c
c258 : 20 33 a5 a6 2b a5 2c 86   ee
c260 : 2d 85 2e a0 00 b1 2d aa   87
c268 : c8 b1 2d d0 f2 a5 2d 18   af
c270 : 69 02 85 2d 90 02 e6 2e   f2
c278 : 20 60 a6 20 13 c2 4c 7b   e5
c280 : e3 a0 03 c8 b1 fb f0 5a   01
c288 : c9 22 d0 09 c8 b1 fb f0   a3
c290 : 51 c9 22 d0 f7 c9 20 d0   58
c298 : ea a6 fb a5 fc 86 8d 85   ce
c2a0 : 8e 18 98 65 fb 85 fb 90   0a
c2a8 : 02 e6 fc a0 00 c8 b1 fb   75
c2b0 : f0 04 c9 20 f0 f7 18 98   79
c2b8 : 65 fb 85 fd a5 fc 69 00   24
c2c0 : 85 fe a0 00 b1 fd 91 fb   36
c2c8 : c8 d0 f9 e6 fc e6 fe a5   a2
c2d0 : 2e c5 fe b0 ef ee 00 d8   df
c2d8 : 20 33 a5 a6 8d a5 8e 4c   a9
c2e0 : 4e c2 a0 00 b1 fb aa c8   ef
c2e8 : b1 fb 4c 4e c2 aa c0 05   02
c2f0 : b0 03 4c 24 c3 88 a9 20   21
c2f8 : d1 fb f0 04 c8 4c 75 c3   d0
c300 : c8 8a 4c 58 c3 a9 c7 48   65
c308 : a9 3c 48 4c 73 00 20 13   49
c310 : c2 a6 2b a5 2c 86 fb 85   97
c318 : fc a0 01 b1 fb d0 03 4c   c6
c320 : 58 c2 a0 03 c8 b1 fb d0   0d
c328 : 03 4c c7 c3 c9 22 d0 09   bf
c330 : c8 b1 fb f0 f4 c9 22 d0   b5
c338 : f7 aa 10 e8 c9 a3 f0 e4   ed
c340 : c9 a6 f0 e0 c9 a4 f0 a5   85
c348 : c9 a7 f0 a1 c9 a9 f0 9d   3e
c350 : c9 af f0 99 c9 b0 f0 95   71
c358 : c9 b4 b0 0c c9 aa 90 08   6d
c360 : c9 af f0 04 c9 b0 d0 bc   9c
c368 : c8 b1 fb f0 5a c9 3a f0   e4
c370 : b3 c9 20 f0 af a6 fb a5   99
c378 : fc 86 8d 85 8e 18 88 98   c8
c380 : 65 fb 85 fb 90 02 e6 fc   72
c388 : 18 a6 38 ca ca 86 fe 86   45
c390 : 25 a5 fb 85 24 69 01 85   d4
c398 : fd 90 02 e6 fe a0 ff b1   93
c3a0 : 24 91 fd 88 d0 f9 b1 24   09
c3a8 : 91 fd c6 25 c6 fe a5 25   d3
c3b0 : c5 fc b0 e9 a0 01 a9 20   56
c3b8 : 91 fb ee 00 d8 20 33 a5   a9
c3c0 : a6 8d a5 8e 4c 15 c3 a0   26
c3c8 : 00 b1 fb aa c8 b1 fb 4c   98
c3d0 : 15 c3 20 13 c2 20 33 a5   77
c3d8 : a6 2b a5 2c 86 fb 85 fc   5b
c3e0 : a0 01 b1 fb d0 03 4c 58   f4
c3e8 : c2 a0 03 c8 b1 fb f0 36   ff
c3f0 : c9 22 d0 09 c8 b1 fb f0   0b
c3f8 : 2d c9 22 d0 f7 c9 8f d0   5a
c400 : ea c0 05 b0 2c a0 00 b1   cd
c408 : fb 85 fd c8 b1 fb 85 fe   6d

c410 : a0 00 b1 fd 91 fb c8 d0   9a
c418 : f9 e6 fc e6 fe a5 2e c5   02
c420 : fe b0 ef 4c d5 c3 a0 00   fa
c428 : b1 fb aa c8 b1 fb 4c dc   80
c430 : c3 84 8d a0 00 b1 fb 85   35
c438 : fd c8 b1 fb 85 fe a4 8d   83
c440 : c0 05 90 cc 88 b1 fb c9   da
c448 : 3a f0 f5 c9 20 f0 f1 c8   94
c450 : a9 00 91 fb 38 98 65 fb   b3
c458 : 85 fb 90 b4 e6 fc b0 b0   10
c460 : 20 df c4 84 fd 85 fe 20   7a
c468 : df c4 84 8d 85 8e 20 13   f0
c470 : c2 a6 2b a5 2c 86 fb 85   f7
c478 : fc a0 00 b1 fb aa c8 b1   96
c480 : fb f0 59 d0 06 a0 01 b1   31
c488 : fb d0 ea a0 03 b1 fb c5   f3
c490 : fe 90 f2 d0 07 88 b1 fb   20
c498 : c5 fd 90 e9 a6 fb a5 fc   98
c4a0 : 86 8b 85 8c a0 00 b1 8b   c7
c4a8 : aa c8 b1 8b d0 08 f0 19   d7
c4b0 : a0 01 b1 8b d0 ea a0 03   9b
c4b8 : b1 8b c5 8e 90 f2 d0 09   68
c4c0 : 88 b1 8b c5 8d 90 e9 f0   a3
c4c8 : e7 a0 00 b1 8b 91 fb c8   fc
c4d0 : d0 f9 e6 fc e6 8c a5 2e   bc
c4d8 : c5 8c b0 ef 4c 58 c2 20   e0
c4e0 : fd ae 20 8a ad 4c f7 b7   1a
c4e8 : 20 fd ae a2 00 20 73 00   d6
c4f0 : f0 0a 48 68 9d 10 02 e8   39
c4f8 : e0 30 90 f1 a9 00 9d 10   84
c500 : 02 a6 2b a5 2c 86 fb 85   c7
c508 : fc a0 01 b1 fb d0 0d 68   16
c510 : 48 60 a0 00 b1 fb aa c8   e8
c518 : b1 fb d0 e9 a0 03 c8 b1   e1
c520 : fb f0 ef cd 10 02 d0 f6   8b
c528 : 18 98 84 8d 65 fb 85 fd   a7
c530 : a5 fc 69 00 85 fe a0 00   81
c538 : c8 b9 10 02 f0 11 c9 3f   5e
c540 : f0 f6 b1 fd 48 68 d9 10   27
c548 : 02 f0 ed a4 8d d0 cf a0   b2
c550 : 02 b1 fb aa c8 b1 fb 20   c9
c558 : cd bd a9 0d 20 d2 ff d0   4a
c560 : b1 a9 20 d0 02 a9 3a 85   69
c568 : fd a6 2b a5 2c 86 fb 85   2a
c570 : fc a0 01 b1 fb d0 03 4c   1e
c578 : 58 c2 a0 04 b1 fb c5 fd   e8
c580 : d0 04 49 1a 91 fb a0 00   63
c588 : b1 fb aa c8 b1 fb 4c 6d   01
c590 : c5 4c 58 c2 20 13 c2 20   d0
c598 : 05 c3 86 41 85 42 20 9e   71
c5a0 : c7 20 93 c7 86 3b 85 3c   26
c5a8 : e4 41 e5 42 b0 03 4c 48   d3
c5b0 : b2 20 93 c7 86 3f 85 40   49
c5b8 : a0 01 b1 43 f0 d3 a9 7f   01
c5c0 : 85 15 85 14 20 05 c3 20   2d
c5c8 : 8e a6 b0 2f a0 00 b1 8b   a3
c5d0 : aa c8 b1 fb f0 bb 48 c8   6a
c5d8 : a5 3b 91 22 c8 a5 3c b1   d1
c5e0 : 24 86 43 68 85 44 a5 3b   ad
c5e8 : 18 65 3f 85 3b a5 3c 65   d0
c5f0 : 40 85 3c 90 d7 e6 7a d0   54
c5f8 : 02 e6 7b a0 00 b1 7a d0   79
c600 : 13 a0 02 b1 7a f0 c5 a5   ac
c608 : 7a 18 69 05 85 7a 90 eb   d0
c610 : e6 7b b0 e7 c9 22 d0 0b   e4
c618 : 20 73 00 c9 00 f0 e2 c9   d2
```




```
c620 : 22 d0 f5 c9 89 f0 17 c9   71
c628 : 8a f0 13 c9 8d f0 0f c9   58
c630 : a7 f0 0b c9 cb d0 be 20   ca
c638 : 73 00 c9 a4 d0 bf ee 00   79
c640 : d8 20 73 00 b0 b7 84 14   08
c648 : 84 15 90 31 aa a5 15 85   d8
c650 : 22 c9 19 b0 a8 a5 14 0a   cf
c658 : 45 22 0a a5 22 65 14 85   8e
c660 : 14 a5 22 65 15 85 15 06   5a
c668 : 14 85 15 8a 65 14 85 14   0b
c670 : 90 02 e6 15 c8 b1 7a c9   f5
c678 : 20 f0 f9 c9 3a a5 14 c5   75
c680 : 41 a5 15 e5 42 90 48 84   69
c688 : ce 20 3c c7 38 a2 90 20   89
c690 : 49 bc 20 dd bd a0 ff c8   6d
c698 : b9 00 01 d0 fa a5 7a a6   c0
c6a0 : 7b 85 5f 86 60 38 98 e5   7c
c6a8 : ce 30 26 f0 6a 85 ce a5   8f
c6b0 : 2d 85 5a 18 65 ce 85 58   cd
c6b8 : a5 2e 85 5b 69 00 85 59   a0
c6c0 : c5 38 90 06 a5 58 c5 37   29
c6c8 : b0 6d 20 bf a3 f0 48 90   33
c6d0 : 5a 49 ff a8 c8 a2 00 a5   d1
c6d8 : 5f c5 2d a5 60 e5 2e b0   69
c6e0 : 36 93 11 11 11 11 21 21   a7
c6e8 : cb 50 49 42 4f 4f 46 54   a7
c6f0 : 21 ce 50 44 4c 46 4f 49   dc
c6f8 : 42 56 51 55 0d 11 44 21   11
c700 : 32 3a 39 35 21 cc 50 46   8b
c708 : 4d 4f 11 1d 1d 0d 00 e6   ed
c710 : 5f d0 c4 e6 60 d0 c0 a0   b6
c718 : 00 b9 00 01 f0 05 91 7a   88
c720 : c8 d0 f6 20 9e c7 20 73   a1
c728 : 00 90 fb a0 00 c9 2c d0   24
c730 : 02 a9 89 aa 4c ff c5 a2   e0
c738 : 10 4c 3a a4 02 a0 00 b9   2a
c740 : e1 c6 f0 18 48 29 7f c9   23
c748 : 20 90 06 68 e9 01 4c 52   bb
c750 : c7 68 20 d2 ff c8 d0 e7   07
c758 : 85 22 86 23 a0 02 84 d3   c8
c760 : 20 7b a4 b1 22 c5 14 c8   cf
c768 : b1 22 e5 15 b0 51 a0 00   5e
c770 : b1 22 aa c8 b1 22 86 22   80
c778 : 85 23 b1 22 f0 41 a5 63   b6
c780 : 65 3f 85 63 a5 62 65 40   d6
c788 : 85 62 b0 04 c9 fc 90 cc   4b
c790 : 4c 48 b2 20 fd ae 20 6b   5e
c798 : a9 a6 14 a5 15 60 20 33   89
c7a0 : a5 18 a5 22 69 02 85 2d   16
c7a8 : a5 23 69 00 85 2e a5 41   1c
c7b0 : a6 42 85 14 86 15 20 13   13
c7b8 : a6 a5 5f 85 43 86 44 60   f4
c7c0 : 20 13 c2 a6 2b a5 2c 86   8d
c7c8 : fb 85 fc ee 00 d8 8a 18   c4
c7d0 : 69 03 85 f7 a5 fc 69 00   03
c7d8 : 85 f8 a0 01 b1 fb d0 03   66
c7e0 : 4c 58 c2 a0 00 84 02 e6   17
c7e8 : f7 d0 02 e6 f8 b1 f7 f0   84
c7f0 : 37 c9 22 d0 1d e6 02 e6   8d
c7f8 : f7 d0 02 e6 f8 b1 f7 f0   94
c800 : 27 c9 22 f0 e0 d0 f0 a0   4c
c808 : 00 b1 fb aa c8 b1 fb 4c   d8
c810 : c7 c7 c9 89 f0 f1 c9 8a   39
c818 : f0 ed c9 8e f0 e9 c9 8f   e8
c820 : f0 e5 c9 a7 d0 bf f0 df   f9
c828 : a0 00 b1 fb 85 fd c8 b1   83
c830 : fb 85 fe c8 b1 fd 85 f9   db
c838 : c8 b1 fd 85 fa 38 a0 00   fd
c840 : b1 fd e5 fb c8 b1 fd e5   c6
c848 : fc d0 bc a6 2b a5 2c 86   4e
c850 : f7 85 f8 a0 01 b1 f7 d0   7b
c858 : 03 4c b7 c8 18 a5 f7 69   ea
c860 : 03 85 7a a5 f8 69 00 85   5f
c868 : 7b a0 00 e6 7a d0 02 e6   14
c870 : 7b b1 7a f0 37 c9 22 d0   6c
c878 : 0e e6 7a d0 02 e6 7b b1   5b
c880 : 7a f0 29 c9 22 d0 f2 c9   fe
c888 : 89 f0 0c c9 8a f0 08 c9   a9
c890 : 8d f0 04 c9 a7 d0 d4 20   64
c898 : 73 00 20 6b a9 a5 15 c5   28
c8a0 : fa d0 c6 a5 14 c5 f9 d0   62
c8a8 : c0 4c 07 c8 a0 00 b1 f7   2a
c8b0 : aa c8 b1 f7 4c 4f c8 ee   6a
c8b8 : 00 04 a0 00 18 a5 fd 69   5c
c8c0 : ff 85 8b a5 fe 69 ff 85   60
c8c8 : 8c a5 02 f0 04 a9 22 10   fc
c8d0 : 02 a9 3a 91 8b c8 c9 22   d2
c8d8 : f0 f7 18 a5 fd 69 04 85   c5
c8e0 : 24 a5 fe 69 00 85 25 e6   52
c8e8 : 8b d0 02 e6 8c a4 02 f0   11
c8f0 : 04 c6 02 f0 f2 b1 24 91   66
c8f8 : 8b c8 d0 f9 e6 8c e6 25   13
c900 : a5 2e c5 25 b0 ef a5 fb   eb
c908 : 69 04 85 8b a5 fc 69 00   2e
c910 : 85 8c b1 8b f0 0a e6 8b   cb
c918 : d0 f8 e6 8c d0 f4 f0 f2   0e
c920 : a0 00 18 a5 8b 69 01 91   a6
c928 : fb c8 a5 8c 69 00 91 fb   57
c930 : c8 b1 8b d0 03 4c 58 c2   47
c938 : 20 33 a5 a6 fb a5 fc 4c   a9
c940 : c7 c7 ad 14 03 ac 15 03   c9
c948 : 8d 0d ca 8c 0e ca a9 5b   35
c950 : a0 c9 78 8d 14 03 8c 15   5a
c958 : 03 58 60 a0 00 b9 7b c9   03
c960 : f0 42 29 3f 99 00 04 a9   a1
c968 : 63 99 50 04 ad 86 02 99   77
c970 : 00 d8 99 28 d8 99 50 d8   95
c978 : c8 d0 e2 49 52 51 20 20   fb
c980 : 42 52 4b 20 20 4e 4d 49   fe
c988 : 20 20 20 50 43 20 20 53   27
c990 : 56 20 42 44 49 5a 43 20   c4
c998 : 41 43 20 58 52 20 59 52   be
c9a0 : 20 53 50 00 ad 0e ca 20   35
c9a8 : 16 ca ad 0d ca 20 16 ca   cc
c9b0 : ad 17 03 20 0f ca ad 16   d8
c9b8 : 03 20 16 ca ad 19 03 20   9a
c9c0 : 0f ca ad 18 03 20 16 ca   c2
c9c8 : bd 06 01 20 0f ca bd 05   15
c9d0 : 01 20 16 ca a9 20 20 2b   33
c9d8 : ca a9 08 85 a4 bd 04 01   74
c9e0 : 85 a5 06 a5 90 04 a9 2a   92
c9e8 : d0 02 a9 2e 20 2b ca c6   fe
c9f0 : a4 d0 ef bd 03 01 20 0f   87
c9f8 : ca bd 02 01 20 0f ca bd   63
ca00 : 01 01 20 0f ca 8a 18 69   a0
ca08 : 06 20 0f ca 4c 31 ea 48   c6
ca10 : a9 20 20 2b ca 68 48 4a   dc
ca18 : 4a 4a 4a 20 1f ca 68 29   5a
ca20 : 0f c9 0a 90 04 e9 09 d0   fe
ca28 : 02 09 30 99 00 04 c8 60   f2
ca30 : ad 0d ca ac 0e ca 4c 52   b9
ca38 : c9 a6 2d a5 2e 86 fb 85   66
ca40 : fc 38 a5 fb e5 2f a5 fc   a9
ca48 : e5 30 b0 55 a0 00 84 d3   e0
ca50 : b1 fb 85 fd 29 7f aa c8   eb
ca58 : b1 fb 85 fe 29 7f a8 20   b9
ca60 : ec ca a5 fd 05 fe 30 3a   58
ca68 : a2 00 20 f6 ca 18 a5 fb   ed
ca70 : 69 02 85 22 a5 fc 69 00   68
ca78 : 85 23 20 a6 bb 20 dd bd   1b
ca80 : 20 1e ab a9 0d 20 d2 ff   ec
ca88 : ad 8d 02 f0 08 c9 01 f0   4f
ca90 : f7 c9 02 f0 0c 18 a5 fb   1b
ca98 : 69 07 aa a5 fc 69 00 90   20
caa0 : 9c 60 a5 fd 30 2a a2 24   bd
caa8 : a0 3d 20 ec ca a9 22 20   4f
cab0 : d2 ff a0 04 b1 fb 85 fe   39
cab8 : 88 b1 fb 85 fd 88 b1 fb   ab
cac0 : 85 8d a0 00 c4 8d b0 bb   27
cac8 : b1 fd 20 d2 ff c8 d0 f4   4e
cad0 : a5 fe 10 b4 a2 25 20 f6   51
cad8 : ca 18 a5 fb 69 02 85 64   1d
cae0 : a5 fc 69 00 85 65 20 61   24
cae8 : af 4c 7d ca 8a f0 03 20   f2
caf0 : d2 ff 98 4c d2 ff a0 3d   9c
caf8 : d0 f2 c2 c1 d3 c9 c3 20   05
cb00 : 2b 20 cb c5 d2 ce c1 cc   2b
cb08 : 20 4a 45 54 5a 54 20 49   85
cb10 : 4d 20 d2 c1 cd 00 a0 00   b9
cb18 : a9 a0 84 fb 85 fc a9 e0   5a
cb20 : 84 fd 85 fe a2 20 b1 fb   ce
cb28 : 91 fb b1 fd 91 fd c8 d0   b1
cb30 : f5 e6 fc e6 fe ca d0 ee   1c
cb38 : b9 fa ca f0 06 20 d2 ff   ec
cb40 : c8 d0 f5 60 d0 d2 c7 2d   17
cb48 : cb 4f 4c 4c 49 53 49 4f   4a
cb50 : 4e 2c 20 c1 45 4e 44 45   57
cb58 : 52 55 4e 47 20 4e 49 43   f1
cb60 : 48 54 20 4d 4f 45 47 4c   59
cb68 : 49 43 48 21 00 a9 6c a0   c9
cb70 : 00 c5 2e b0 0b b9 44 cb   1c
cb78 : f0 c9 20 d2 ff c8 d0 f5   25
cb80 : 84 37 c5 38 b0 02 85 38   ba
cb88 : a2 10 a9 d0 84 fb 85 fc   ef
cb90 : a9 70 84 fd 85 fe ad 0e   75
cb98 : dc 29 fe 8d 0e dc a5 01   da
cba0 : 29 fb 85 01 b1 fb 91 fd   86
cba8 : c8 d0 f9 e6 fc e6 fe ca   cc
cbb0 : d0 f2 a5 01 09 04 85 01   4c
cbb8 : ad 0e dc 09 01 8d 0e dc   33
cbc0 : ad 00 dd 29 fc 09 02 8d   45
cbc8 : 00 dd a9 bc 8d 18 d0 a9   e9
cbd0 : 6c 8d 88 02 b9 df cb f0   11
cbd8 : 35 20 d2 ff c8 d0 f5 93   e4
cbe0 : 11 11 11 0e 24 36 43 30   e1
cbe8 : 30 3a c2 49 4c 44 53 43   ca
cbf0 : 48 49 52 4d 0d 11 24 37   73
cbf8 : 30 30 30 3a da 45 49 43   17
cc00 : 48 45 4e 47 45 4e 45 52   e8
cc08 : 41 54 4f 52 0d 00 4c 58   44
cc10 : c2 a9 93 20 d2 ff a0 00   3f
cc18 : b9 26 cc d0 03 4c a6 cc   f8
cc20 : 20 d2 ff c8 d0 f2 13 11   d5
cc28 : 11 11 4d 49 54 20 12 c3   54
cc30 : d5 d2 d3 cf d2 92 20 20   e0
cc38 : 20 da 45 49 43 48 45 4e   68
cc40 : 20 45 49 4e 53 54 45 4c   a4
cc48 : 4c 45 4e 2c 0d 4d 49 54   59
cc50 : 20 12 d2 c5 d4 d5 d2 ce   cb
cc58 : 92 20 20 20 57 41 45 48   2c
cc60 : 4c 45 4e 2e 0d 1d 1d 1d   11
cc68 : 1d 12 46 37 92 3a 20 20   c3
cc70 : 20 20 20 20 da 45 49 43   30
cc78 : 48 45 4e 53 41 54 5a 0d   9b
cc80 : 1d 1d 1d 1d 12 d2 d5 ce   c4
cc88 : 2f d3 d4 cf d0 92 3a c1   de
cc90 : 42 42 52 55 43 48 2e 11   84
cc98 : 0d 1d 1d 1d 1d 1d 1d 1d   88
cca0 : 1d 1d 1d 1d 1d 00 a8 84   b4
cca8 : 8d 98 c9 20 b0 04 a9 20   0a
ccb0 : d0 07 aa 10 04 c9 a0 90   e3
ccb8 : f5 20 d2 ff e6 8d a5 8d   fe
ccc0 : 29 0f d0 09 a9 0d 20 d2   ef
ccc8 : ff a9 0b 85 d3 c8 d0 d9   8a
ccd0 : b9 db cc f0 1b 20 d2 ff   c6
ccd8 : c8 d0 f5 13 11 11 11 11   e8
cce0 : 11 11 11 11 1d 1d 1d 1d   4a
cce8 : 1d 1d 1d 1d 1d 1d 1d 00   ae
ccf0 : 85 fb a9 00 85 fc 20 41   20
ccf8 : cf c9 1d d0 25 e6 fc e6   58
cd00 : d3 a6 fc e0 10 90 ef c6   54
cd08 : d3 c6 fc a4 fb c0 0f b0   75
cd10 : e5 e6 fb a9 0d 20 d2 ff   b9
cd18 : a9 00 85 fc 09 0b 85 d3   69
cd20 : d0 d0 c9 13 d0 04 a0 00   dd
cd28 : f0 a6 c9 9d d0 0a a6 fc   83
cd30 : f0 c4 c6 fc c6 d3 d0 be   9f
cd38 : c9 11 d0 0d a4 fb c0 0f   ab
cd40 : b0 b4 20 d2 ff e6 fb d0   75
cd48 : ad c9 91 d0 0b a4 fb f0   00
cd50 : a5 20 d2 ff c6 fb d0 9e   87
cd58 : c9 03 d0 03 4c 58 c2 c9   5d
cd60 : 0d f0 0f c9 88 d0 08 ad   6d
cd68 : 18 d0 49 02 8d 18 d0 4c   f0
cd70 : f6 cc a5 fc a4 fb f0 eb   9b
cd78 : 18 69 10 88 d0 fa 48 29   b2
cd80 : 80 85 fd 68 29 7f c9 20   45
cd88 : 90 d9 c9 40 90 02 29 3f   bb
cd90 : a4 fd f0 02 09 40 85 fb   50
cd98 : ad 18 d0 29 07 4a 09 0e   ae
cda0 : 85 fc 06 fb 26 fc 06 fb   ff
cda8 : 26 fc 06 fb 26 fc a9 07   4c
cdb0 : 85 8d a0 00 b9 c2 cd d0   ae
cdb8 : 03 4c 38 ce 20 d2 ff c8   f3
cdc0 : d0 f2 13 11 11 11 11 11   f0
cdc8 : 11 11 11 11 11 11 11 1d   e0
cdd0 : 1d 1d 1d 1d 1d 1d 1d 1d   d0
cdd8 : 1d 1d 1d 1d 1d b0 c0 c0   4b
cde0 : c0 c0 c0 c0 c0 c0 ae 11   37
cde8 : 9d dd 11 9d dd 11 9d dd   04
cdf0 : 11 9d dd 11 9d dd 11 9d   b2
cdf8 : dd 11 9d dd 11 9d dd 11   18
ce00 : 9d dd 9d 9d 9d 9d 9d 9d   1f
ce08 : 9d 9d 9d 9d ad c0 c0 c0   f4
ce10 : c0 c0 c0 c0 c0 bd 9d 9d   24
ce18 : 9d 9d 9d 9d 9d 9d 9d 91   ff
ce20 : 9d dd 91 9d dd 91 9d dd   60
ce28 : 91 9d dd 91 9d dd 91 9d   7c
ce30 : dd 91 9d dd 91 9d dd 00   76
ce38 : a5 8d 49 07 a8 b1 fb 85   ea
ce40 : 8e a9 0f 85 d3 a0 08 a9   cd
ce48 : 20 06 8e 90 02 09 0a 20   f2
ce50 : d2 ff 88 d0 f2 a9 0d 20   4f
ce58 : d2 ff c6 8d 10 da b9 69   1f
ce60 : ce f0 24 20 d2 ff c8 d0   a5
ce68 : f5 13 11 11 11 11 11 11   4d
ce70 : 11 11 11 11 11 11 11 1d   88
ce78 : 1d 1d 1d 1d 1d 1d 1d 1d   78
ce80 : 1d 1d 1d 1d 1d 1d 00 85   dd
ce88 : f8 85 f7 20 41 cf c9 1d   39
ce90 : d0 0d a6 f7 e0 07 b0 f3   80
ce98 : e6 f7 20 d2 ff d0 ec c9   aa
cea0 : 9d d0 0b a6 f7 f0 e4 c6   65
cea8 : f7 20 d2 ff d0 dd c9 11   a9
ceb0 : d0 0d a4 f8 c0 07 b0 d3   fe
ceb8 : 20 d2 ff e6 f8 d0 cc c9   fb
cec0 : 91 d0 0b a4 f8 f0 c4 c6   c8
cec8 : f8 20 d2 ff d0 bd c9 20   e7
ced0 : f0 0b c9 2a f0 2b c9 0d   a7
ced8 : d0 b1 4c 16 cc a6 f7 e0   fa
cee0 : 08 b0 f3 20 d2 ff a9 80   16
cee8 : a6 f7 f0 02 4a ca d0 fc   3e
cef0 : 49 ff 85 fd a4 f8 b1 fb   2b
cef8 : 25 fd 91 fb e6 f7 4c 8b   76
cf00 : ce a6 f7 e0 08 b0 d6 20   dd
```

Listing 1. »HELPS«. Bitte mit dem MSE (Seite 159) eingeben.

```
cf08 : d2 ff a9 80 a6 f7 f0 02  46
cf10 : 4a ca d0 fc a4 f8 11 fb  e1
cf18 : d0 e0 a9 04 a0 97 8d 88  51
cf20 : 02 8c 00 dd 20 5b ff a9  54
cf28 : 05 20 d2 ff a9 08 20 d2  f3
cf30 : ff a9 0e 20 d2 ff a9 0b  75
cf38 : 8d 20 d0 8d 21 d0 4c 58  36
cf40 : c2 46 cc 46 cf 20 e4 ff  b3
cf48 : f0 f7 48 e6 cc a4 d3 b1  c7
cf50 : d1 29 7f 91 d1 68 60 20  ea
cf58 : fd ae 20 d4 e1 a2 01 a0  c7
cf60 : 80 86 b9 ca 86 8d a9 70  47
cf68 : 85 8e a9 8d 4c d8 ff 20  1c
cf70 : 20 ca 4f 48 41 4e 4e 45  1d
cf78 : 53 20 cd 4f 43 4b 45 4e  79
cf80 : 48 41 55 50 54 20 20 cb  27
cf88 : 4f 45 4c 4e 43 20 31 39  c3
cf90 : 38 34 20 d4 e1 a2 01 86  c9
cf98 : b9 ca 86 8e a0 d0 a9 c0  e2
cfa0 : 85 8f a9 8e 4c d8 ff 00  b4
cfa8 : 00 00 00 00 00 00 00 00  a9
cfb0 : 00 00 00 00 00 00 00 00  b1
cfb8 : 00 00 00 00 00 00 00 00  b9
cfc0 : 00 00 00 00 00 00 00 00  c1
cfc8 : 00 00 00 00 00 00 00 00  c9
cfd0 : 00 00 00 00 00 00 00 00  d1
cfd8 : 00 00 00 00 00 00 00 00  d9
cfe0 : 00 00 00 00 00 00 00 00  e1
cfe8 : 00 00 00 00 00 00 00 00  e9
cff0 : 00 00 00 00 00 00 00 00  f1
cff8 : 00 00 00 00 00 00 00 00  f9
```

Listing 1. (Schluß)

Alte Texte im neuen Kleid

Babylonische Verwirrung herrscht bei den Textformaten verschiedener Textprogramme. Dies macht sich unangenehm bemerkbar, wenn man seine Dokumente grafisch ansprechend – etwa mit dem Printfox – drucken will. Die folgenden kurzen Programme schaffen Abhilfe.

Wem das Programm »Printfox« (siehe Ausgabe 11/87 des 64'er-Magazins) so gut gefällt, daß er damit auch seine alten Texte drucken will, dem wird jetzt geholfen. Mit den vier folgenden Konvertierungsprogrammen (Listing 1 bis 4) können Sie Texte in den Formaten von:
- Textomat nach Printfox
- Mastertext nach Printfox
- Printfox nach Mastertext
- Printfox nach Textomat

umwandeln. Über den Umweg eines Printfox-Textes kann so auch
- Textomat nach Mastertext
- Mastertext nach Textomat

umkopiert werden. Die Bedienung der Programme ist sehr einfach. Nach Start eines der Konvertierprogramme mit RUN muß lediglich der Name des Originaltextes und der Name des Files, unter dem der konvertierte Text gespeichert wird, eingegeben werden. Daraufhin wird eine Wandlung sämtlicher druckbarer Zeichen vorgenommen; Format-Steuerzeichen werden ignoriert. Der Kontrollausdruck am Bildschirm kann mitunter etwas chaotisch aussehen, aber keine Angst, denn bei den Grafikzeichen handelt es sich um Umlaute, die im Textprogramm dann wieder als solche erscheinen.

Beispiel:

Sie haben einen Brief mit Textomat geschrieben und entdecken nun, daß dies mit Printfox wesentlich schöner auszudrucken wäre.
1. Programm »FOX-TEXTO« in Rechner laden.
2. Diskette mit Textomat-Text in das Laufwerk legen.
3. Programm starten und Filenamen eintippen.

Das Programm liest nun aus dem Quelltext (Textomat) Zeichen für Zeichen ein, wandelt nach Bedarf um und schreibt die neuen Zeichen in den Zieltext (Printfox). Parallel dazu erfolgt eine Ausgabe auf dem Bildschirm. Nach der Ende-Meldung kann der neue Text direkt von Printfox eingelesen und weiterverarbeitet werden. Übrigens: Unbedingt darauf achten, daß die Anzahl der freien Blöcke auf der Diskette größer als die Länge des Textes ist, sonst kommt es zu einem DISK FULL ERROR.

Die vier Basic-Listings werden mit dem Checksummer eingegeben.

(Peter Engels/Stefan Willmeroth)

```
100 REM          TEXTO-FOX                        <023>
110 REM                                           <172>
120 REM  01.03.87  BY P.ENGELS                    <148>
130 REM  ---------------------                    <121>
140 REM                                           <202>
150 PRINT CHR$(14)"{CLR,2DOWN}TEXTOMAT ->
    PRINTFOX"CHR$(13)"{DOWN,3SPACE}TEXTKON
    VERTER{2DOWN}"                                <182>
160 INPUT"QUELL-TEXT:";QF$:QF$=QF$+",P,R"  <100>
170 INPUT" ZIEL-TEXT:";ZF$:ZF$=ZF$+",P,W"  <201>
180 OPEN 1,8,3,QF$                                <048>
190 OPEN 2,8,4,ZF$                                <213>
200 PRINT#2,CHR$(84);:REM TEXT-KENNUNG            <082>
210 :                                             <186>
220 GET#1,A$:IF A$=""THEN A$=CHR$(0)              <025>
230 A=ASC(A$):B=0                                 <145>
240 IF(A>64)AND(A<91)THEN B=A+32:GOTO 370:
    REM KLEIN                                     <100>
250 IF(A>192)AND(A<219)THEN B=A-128:GOTO 3
    70:REM GROSS                                  <115>
260 IF(A>31)AND(A<65)THEN B=A:GOTO 370:REM
     ZIFFERN & ZEICHEN                            <094>
270 IF A=160 THEN B=127:REM SHIFT SPACE           <132>
280 IF A=95 THEN B=13:REM ABSATZ                  <197>
290 IF A=169 THEN B=91:REM AE                     <182>
300 IF A=168 THEN B=92:REM OE                     <087>
310 IF A=170 THEN B=93:REM UE                     <176>
320 IF A=165 THEN B=123:REM AE                    <025>
330 IF A=161 THEN B=124:REM OE                    <079>
340 IF A=166 THEN B=125:REM UE                    <002>
350 IF A=167 THEN B=126:REM SZ                    <212>
360 :                                             <082>
370 IF A=95 THEN PRINT CHR$(13);:GOTO 390         <006>
380 PRINT CHR$(A);                                <179>
390 IF ST=2 GOTO 430:REM FILE-ENDE                <176>
400 IF B=0 GOTO 220                               <016>
410 PRINT#2,CHR$(B);:GOTO 220                     <188>
420 :                                             <142>
430 PRINT#2,CHR$(0);CHR$(0)                       <166>
440 CLOSE 1:CLOSE 2                               <008>
450 PRINT:PRINT:PRINT"{RVSON,SPACE}TEXT IS
    T KONVERTIERT !!{RVOFF}"                      <170>
```

Listing 1. »TEXTO-FOX« Textomat wird zu Printfox. Bitte mit dem Checksummer (Seite 159) eingeben.

```
100 REM          MASTER-FOX                       <005>
110 REM                                           <172>
120 REM   01.03.87  BY  P.ENGELS                  <148>
130 REM   ---------------------                   <166>
140 REM                                           <202>
150 PRINT CHR$(14)"{CLR,2DOWN}MASTERTEXT -
    > PRINTFOX"CHR$(13)"{DOWN,SPACE}TEXTKO
    NVERTER{2DOWN}"                               <156>
160 INPUT"QUELL-TEXT:";QF$:QF$=QF$+",P,R"  <100>
```

```
170 INPUT" ZIEL-TEXT:";ZF$:ZF$=ZF$+",P,W"  <201>
180 OPEN 1,8,3,QF$                          <048>
190 OPEN 2,8,4,ZF$:PRINT#2,CHR$(84);:REM T
    EXT-KENNUNG                             <018>
200 PRINT:PRINT                             <152>
210 GET#1,A$:GET#1,A$:REM START ADRESSE UE
    BERLESEN                                <165>
220 GET#1,A$: IF A$=""THEN A$=CHR$(0)       <025>
230 A=ASC(A$):B=0                           <145>
240 IF(A<27)THEN B=A+96:GOTO 360:REM KLEIN  <001>
250 IF(A>31)AND(A<91)THEN B=A:GOTO 360:REM
     ZIFFERN,ZEICHEN,GROSSBUCHST.           <068>
260 IF A=160 THEN B=127:REM SHIFT SPACE     <122>
270 IF A=142 THEN B=13:REM ABSATZ           <110>
280 IF A=91 THEN B=91:REM AE                <093>
290 IF A=92 THEN B=92:REM OE                <050>
300 IF A=93 THEN B=93:REM UE                <006>
310 IF A=27 THEN B=123:REM AE               <191>
320 IF A=28 THEN B=124:REM OE               <033>
330 IF A=29 THEN B=125:REM UE               <126>
340 IF A=30 THEN B=126:REM SZ               <138>
350 :                                       <072>
360 PRINT CHR$(B);                          <191>
370 IF ST=2 GOTO 420:REM FILE-ENDE          <154>
380 IF B=0 GOTO 220                         <252>
390 PRINT#2,CHR$(B);:IF B=13 GOTO 460:REM
    SPACES UEBERLESEN                       <215>
400 GOTO 220                                <114>
410 :                                       <132>
420 PRINT#2,CHR$(0);CHR$(0):REM ENDEKENNUN
    G                                       <049>
430 CLOSE 1:CLOSE 2                         <254>
440 PRINT:PRINT:PRINT"{RVSON}TEXT IST KONV
    ERTIERT !!{RVOFF}":END                  <061>
450 :                                       <172>
460 GET#1,A$: IF A$=""THEN A$=CHR$(00):GOTO
     460                                    <050>
470 IF A$=CHR$(32)THEN 460:SPACES UEBERLES
    EN                                      <203>
480 GOTO 230                                <210>
```

Listing 2. »MASTER-FOX« Master-Text wird zu Printfox. Bitte mit dem Checksummer (Seite 159) eingeben.

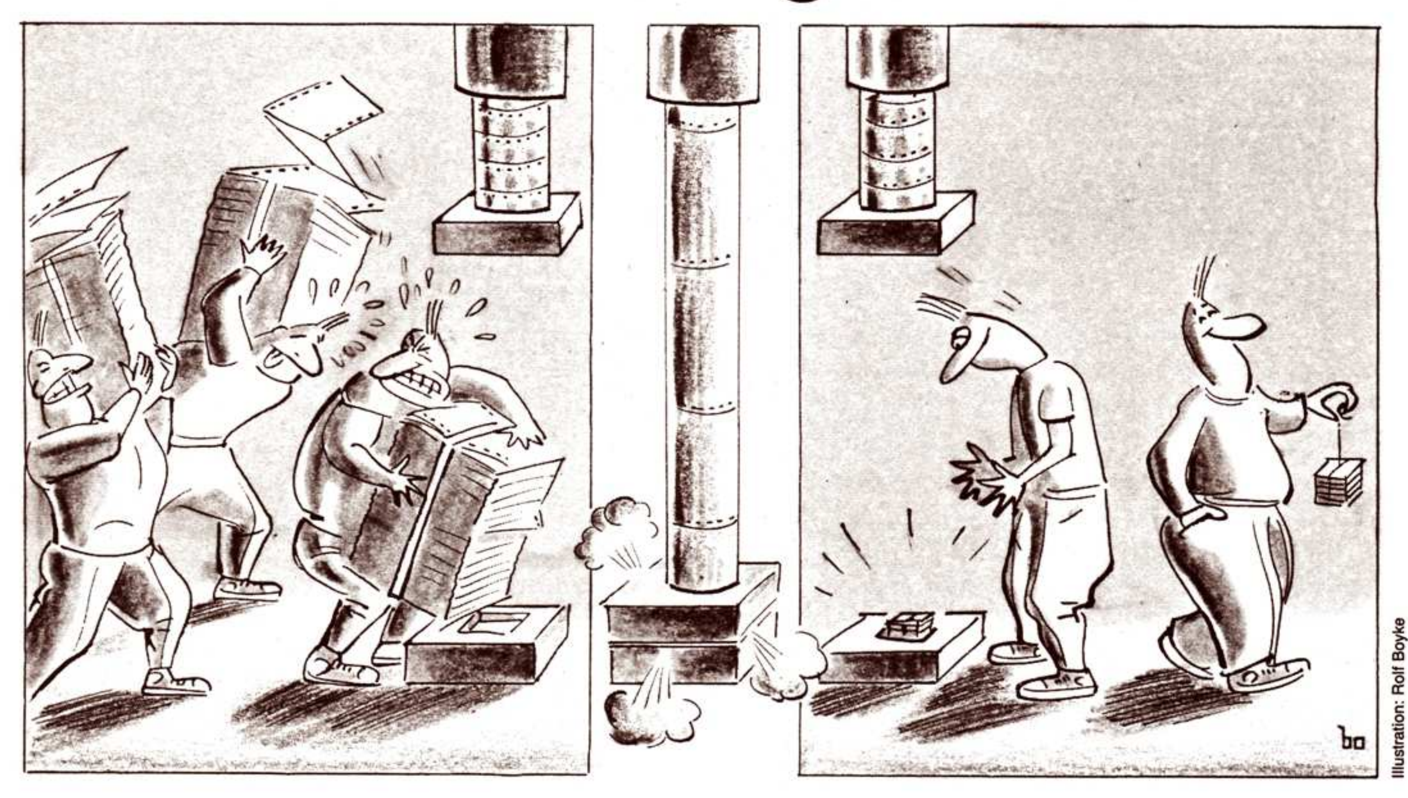
```
100 REM        FOX-TEXTO                    <008>
110 REM                                     <172>
120 REM   01.03.87   BY P.ENGELS            <148>
130 REM   ---------------------             <121>
140 REM                                     <202>
150 PRINT CHR$(14)"{CLR,2DOWN}PRINTFOX ->
    TEXTOMAT"CHR$(13)"{DOWN,2SPACE}TEXTKON
    VERTER{2DOWN}"                          <245>
160 INPUT"QUELL-TEXT:  ";QF$:QF$=QF$+",P,R" <100>
170 INPUT" ZIEL-TEXT:  ";ZF$:ZF$=ZF$+",P,W" <201>
180 OPEN 1,8,3,QF$                          <048>
190 OPEN 2,8,4,ZF$                          <213>
200 PRINT:PRINT:GET#1,A$:REM TEXTKENNUNG    <153>
210 GET#1,A$: IF A$=""THEN A$=CHR$(0)       <015>
220 A=ASC(A$):B=0                           <135>
230 IF(A>96)AND(A<123)THEN B=A-32:GOTO 370
    :REM KLEIN                              <085>
240 IF(A>64)AND(A<91)THEN B=A+128:GOTO 370
    :REM GROSS                              <180>
250 IF(A>31)AND(A<65)THEN B=A:GOTO 370:REM
     ZIFFERN & ZEICHEN                      <082>
260 IF A=2 THEN GOTO 470:REM FORMATZEILE I
    GNORIEREN                               <162>
270 IF A=160 THEN B=127:REM SHIFT SPACE     <132>
280 IF A=13 THEN B=95:REM ABSATZ            <115>
290 IF A=91 THEN B=169:REM AE               <139>
300 IF A=92 THEN B=168:REM OE               <204>
310 IF A=93 THEN B=170:REM UE               <161>
320 IF A=123 THEN B=165:REM AE              <151>
330 IF A=124 THEN B=161:REM OE              <176>
340 IF A=125 THEN B=166:REM UE              <224>
350 IF A=126 THEN B=167:REM SZ              <179>
360 :                                       <082>
370 IF(A>31)THEN PRINT CHR$(A);             <062>
380 IF A=13 THEN PRINT CHR$(A);             <113>
390 IF ST=2 GOTO 430:REM FILE-ENDE          <176>
400 IF B=0 GOTO 210                         <015>
410 PRINT#2,CHR$(B);:GOTO 210               <156>
420 :                                       <142>
430 PRINT#2                                 <163>
440 CLOSE 1:CLOSE 2                         <008>
450 PRINT:PRINT:PRINT"{RVSON}TEXT IST KONV
    ERTIERT !!{RVOFF}":END                  <071>
460 :                                       <182>
470 GET#1,A$: IF ASC(A$)<>13 GOTO 470:ZEILE
     UEBERLESEN                             <229>
480 GOTO 210                                <178>
```

Listing 3. »FOX-TEXTO« Printfox wird zu Textomat. Bitte mit dem Checksummer (Seite 159) eingeben.

```
100 REM          FOX-MASTER                 <075>
110 REM                                     <172>
120 REM    02.03.87   BY  P.ENGELS          <164>
130 REM    ----------------------           <166>
140 REM                                     <202>
150 PRINT CHR$(14)"{CLR,2DOWN}PRINTFOX ->
    MASTERTEXT"CHR$(13)"{DOWN,SPACE}TEXTKO
    NVERTER{2DOWN}"                         <201>
160 INPUT"QUELL-TEXT:";QF$:QF$=QF$+",P,R"   <100>
170 INPUT" ZIEL-TEXT:";ZF$:ZF$=ZF$+",P,W"   <201>
180 OPEN 1,8,3,QF$                          <048>
190 OPEN 2,8,4,ZF$:PRINT#2,CHR$(0)CHR$(08)
    ;:REM START-KENNUNG                     <166>
200 PRINT:PRINT                             <152>
210 GET#1,A$:REM TEXT-KENNUNG UEBERLESEN    <114>
220 GET#1,A$: IF A$=""THEN A$=CHR$(0)       <025>
230 A=ASC(A$):B=0                           <145>
240 IF(A>96)AND(A<123)THEN B=A-96:GOTO 370
    :REM KLEIN                              <227>
250 IF(A>31)AND(A<91)THEN B=A:GOTO 370:REM
     ZIFFERN,ZEICHEN,GROSSBUCHST.           <196>
260 IF A=2 THEN GOTO 540:REM FORMATZEILE I
    GNORIEREN                               <157>
270 IF A=127 THEN B=32:REM SHIFT SPACE      <078>
280 IF A=13 THEN B=142:REM ABSATZ           <111>
290 IF A=91 THEN B=91:REM AE                <103>
300 IF A=92 THEN B=92:REM OE                <060>
310 IF A=93 THEN B=93:REM UE                <016>
320 IF A=123 THEN B=27:REM AE               <248>
330 IF A=124 THEN B=28:REM OE               <154>
340 IF A=125 THEN B=29:REM UE               <055>
350 IF A=126 THEN B=30:REM SZ               <078>
360 :                                       <082>
370 IF(A>31)OR(A=13)THEN PRINT CHR$(A);     <183>
380 IF ST=2 GOTO 440:REM FILE-ENDE          <168>
390 IF B=0 THEN 220                         <141>
400 PRINT#2,CHR$(B);                        <009>
410 SZ=SZ+1:REM SPALTENZAEHLER              <218>
420 IF B=142 THEN GOSUB 480:REM ZEILE AUFF
    UELLEN                                  <061>
430 GOTO 220                                <144>
440 PRINT#2,CHR$(0)CHR$(0):REM ENDE         <171>
450 CLOSE 1:CLOSE 2                         <018>
460 PRINT:PRINT"{DOWN,RVSON}TEXT IST KONVE
    RTIERT !!{RVOFF}":END                   <027>
470 :                                       <192>
480 IF SZ> 79 THEN PRINT#2," ";             <035>
490 IF SZ>159 THEN PRINT#2," ";             <027>
500 IF SZ>319 THEN PRINT#2," ";             <163>
510 IF SZ>399 THEN PRINT#2," ";             <177>
520 FOR Z=SZ TO 79*(INT(SZ/79)+1):PRINT#2,
    " ";:NEXT:SZ=0:RETURN:REM AUFFUELLEN    <006>
530 :                                       <254>
540 GET#1,A$: IF ASC(A$)<>13 GOTO 540:ZEILE
     UEBERLESEN                             <221>
550 GOTO 220                                <010>
```

Listing 4. »FOX-MASTER« Printfox wird zu Mastertext. Bitte mit dem Checksummer (Seite 159) eingeben.

Die Super-Diät für Ihre Programme


Illustration: Rolf Boyke

Einen Programm-Kompressor und Packer vom Feinsten bieten wir Ihnen mit dem »64'er-Packer«. Dank eines ausgefeilten Kompressionsverfahrens kürzt er sogar mehrteilige und sehr lange Programme radikal. Durch frei wählbare Packverfahren ist er äußerst flexibel und jedem »Programm-Monster« gewachsen.

Der 64'er-Packer ist ein sehr leistungsfähiger Mehrpaß-Kompressor mit vielen Kombinations- und Einstellmöglichkeiten, wobei jeder Paß ein für sich lauffähiges Programm liefert. Er enthält ferner mit einem der Vorkompressoren einen »Linker«, der zum Zusammenfügen mehrteiliger Programme fast den ganzen RAM-Speicherplatz zur Verfügung stellt. Lücken zwischen den einzelnen Teilen verschwinden beim Komprimieren fast rückstandslos.

Aus 115 mach 52

So gelang es in der Redaktion beispielsweise, die 115 Blocks lange compilierte Version unseres Basic-Compilers »Hypra-Comp« aus dieser Ausgabe auf sage und schreibe 52 Blöcke zu kürzen. Unser Textverarbeitungsprogramm Master-Text war als einteilige gepackte Version gar nur noch 48 Blöcke lang. Der Kompressor und das eingebaute Fastload stellen zum Packen und Laden den RAM-Bereich von $01E0 bis $FFFF (also mehr als 63,5 KByte!) bereit. Die maximale, vom Vorkompressor verarbeitete Programmlänge beträgt $FE1E Byte.
Um all dies zu ermöglichen, mußte ein kleiner Kompromiß eingegangen werden:

Der 64'er-Packer (Listing 1) muß vor der Inbetriebnahme auf ein 8-KByte-EPROM vom Typ 2764 gebrannt werden. Dieses EPROM muß dann auf einer handelsüblichen EPROM-Karte, die den Bereich ab $8000 adressiert, in den Expansion-Port des C 64 gesteckt werden. Das Listing stellt den (übrigens ungepackten) Packer in direkt brennfähiger Form dar.

Um Ihnen ein angenehmeres Arbeiten zu ermöglichen, wurden in den 64'er-Packer ein Schnellader (ca. 5mal schneller als die Standardbetriebsystem-Routinen) für die zu packenden Programme und ein komfortables Directory-Auswahlmenü integriert. Aus diesen Gründen kann es zu Inkompatibilitäten mit manchen Hardware-Speedern kommen. Mit Speeddos und Dolphin-DOS abeitet der Packer jedoch problemlos zusammen.

Bei der Realisierung der Verfahren und der Dekompressoren beziehungsweise der Abstimmung der Dekompressoren aufeinander, wurde sehr viel Wert darauf gelegt, notfalls etwas mehr Aufwand zu treiben, um sicher zu sein, das entstehende Programm nicht unnötig um 1 Byte verlängert zu haben.

Nachdem Sie das EPROM eingeschaltet haben, kann der 64'er-Packer durch einen Reset (Druck auf den Resettaster oder Einschalten des Rechners) oder auch durch SYS 64738 gestartet werden. Es erscheint dann ein Menü (Bild 1), bei dem man fünf verschiedene Funktionen per Tastendruck anwählen kann:

<**V**>	Vorkompressor
<**D**>	Directory
<**Z**>	Zwischenkompressor
<**E**>	Exit
<**N**>	Nachkompressor



Vor der Beschreibung der verschiedenen Kompressortypen, erhalten Sie zunächst eine Anleitung zum Directory-Auswahlmenü.

Haben Sie einen der drei Kompressoren (über <V>, <Z> oder <N) im Hauptmenü angewählt, so werden Ihnen zunächst die (maximal) 21 ersten Files der eingelegten Diskette angezeigt. Nun stehen folgende Tasten zur Wahl:

<SPACE>
Auswahl des ersten angezeigten Files. Ein ausgewähltes File wird revers dargestellt.

<CURSOR aufwärts>
Scrollen um einen Fileeintrag nach oben

<CURSOR abwärts>
Scrollen um einen Fileeintrag abwärts

<A>
Auswahl aller Files ab dem an erster Stelle angezeigten

<B>
Anzeigen des letzten Files

<C>
Desaktivieren aller Files ab dem an erster Stelle angezeigten

<T>
Anzeigen der (maximal) 21 ersten Files (wie zu Anfang)

<←>
Beenden der Auswahl

Für die ausgewählten Filenamen stehen insgesamt 160 Byte zur Verfügung, wobei ein Eintrag »Namenslänge+1« Byte belegt. Somit können bei einbuchstabigen Namen maximal 80 Files vermerkt werden. Alle angewählten Files werden dann später mit einem Schnellader, der ungefähr die 5fache Geschwindigkeit der Standardbetriebssystem-Routinen besitzt, geladen – das im Directory am weitesten hinten liegende zuerst.

Tritt hierbei ein Fehler auf, kann man durch Drücken von <SPACE> ein erneutes Laden des entsprechenden Files versuchen oder mittels <RUN/STOP> einen Abbruch herbeiführen.

<V> Vorkompressor/Linker
Zuerst erfolgt die Auswahl des beziehungsweise der zu komprimierenden Files wie oben beschrieben. Es dürfen so viele Files angewählt werden, wie in den 160 Byte großen Namensspeicher aufgenommen werden können. Es wird nochmals darauf hingewiesen, daß in der Directory weiter hinten stehende Files zuerst geladen werden. Dies ist jedoch nur bei sehr vielen Files, die zu einem Programm gepackt werden sollen, relevant.

Drei Packer-Verfahren

Danach werden Sie zunächst nach dem Namen des zu erzeugenden, vorkomprimieren Files gefragt und dann nach der Kompressions-Art, wobei die drei Möglichkeiten »A«, »B« oder »C« zur Auswahl stehen:

Typ A
Dieser Typ gilt für alle Maschinenprogramme, beziehungsweise Programme oder Grafikbilder, die keine Basic-Zeilen beinhalten, beziehungsweise auf die Zeropage-Adressen $2D/$2E (Zeiger auf Basic-Programmende/Variablenanfang) angewiesen sind. Die Endadresse eines so gepackten Programms steht nach dem Dekomprimieren in $AE/ $AF.

Typ B
Dieses Packverfahren gilt auch für die Ausnahmen bei Typ »A«. Die Endadresse nach dem Dekomprimieren steht in $AE/ $AF und $2D/$2E. Geeignet ist dieses Verfahren besonders für Maschinenprogramme mit Basic-Start. Programme, die mit diesem Verfahren gepackt werden, sind in der Regel 8 Byte länger als solche vom Typ A.

Typ C
Für Basic-Programme geeignet. Das Verfahren ist ähnlich Typ »B«, jedoch erfolgt der Programmstart nach dem Dekomprimieren mit RUN, statt mit SYS. Die gepackten Programme sind daher in der Regel 11 Byte länger als bei Verwendung von Typ »A«.

Tips zur Vorgehensweise

1. Mehrteilige Programme mit Lader
Natürlich darf ein eventuell vorhandener Lader, der normalerweise die einzelnen Programmteile in den Speicher lädt, nicht mitkomprimiert werden. Sonst würde bei mehrteiligen, komprimierten Programmen ja noch einmal alles nachgeladen. Erfüllt ein solches Ladeprogramm allerdings noch andere Aufgaben, so muß das zu komprimierende Programm entsprechend modifiziert werden (Einfügen der Routinen). Eine andere Möglichkeit besteht darin, alle LOAD-Befehle etc. aus dem Ladeprogramm zu entfernen. Beim Programm »LADER« von Master-Text funktionierte diese Methode beispielsweise einwandfrei.

2. Basic-Programme und Maschinenroutinen
Wollen Sie Basic-Programme und Maschinenroutinen zusammen komprimieren, sollte nicht unbedingt immer Typ »B« oder »C« gewählt werden, wie man vielleicht annehmen könnte. Dazu das folgende Beispiel:

Man möchte ein Basic-Programm zusammen mit Maschinenroutinen, die ab $C000 liegen, packen. Da bei Typ »B« beziehungsweise »C« die Endadresse nach dem Dekomprimieren in $AE/$AF und $2D/ $2E liegt und größer als $C000 ist, würde man bei Start des Basic-Programmes einen »OUT OF MEMORY ERROR« erhalten, da gemäß $2D/



Bild 1. Das Hauptmenü des 64'er-Packers

$2E kein Speicherplatz mehr für die Basic-Variablen vorhanden ist. Diesem kann man mit zwei verschiedenen Methoden begegnen, wobei man allerdings die Endadresse des unkomprimierten Basic-Programmes wissen muß (siehe Adressen 45/ 46 = $2D/ $2E):

a) Die ersten Befehle des Basic-Programmes müssen lauten:

```
1 POKE 45,XXX: POKE 46,YYY: CLR
```

Für »XXX« setze man nun (3stellig, notfalls führende Null(en)!) den Wert ein, den man mit `PRINT PEEK(45)` erhält. Für »YYY« setzt man analog den Wert:

```
PRINT PEEK(46)
```

Dieser muß ebenfalls 3stellig eingegeben werden. Danach darf die Programmlänge auf keinen Fall verändert werden, es sei denn, man setzt danach die neuen PEEK-Werte in Zeile 1 ein;

b) Man schreibt ein kurzes Maschinenprogramm, das man

am besten im freien RAM-Bereich (beispielsweise ab $0334) plaziert und das zuerst aufgerufen wird:

```
LDA #↑<(Basic-Programm Ende) ;Low-Byte
STA $2D
LDA #↑>(Basic-Programm Ende) ;High-Byte
STA $2E
JSR $A659                    ;Basic-Befehl CLR
JMP $A7AE                    ;RUN
```

oder

JMP $C000, falls Maschinenroutinen ab $C000 zuerst aufgerufen werden müssen.

Es ist dann natürlich sinnvoller, Typ »A« zu verwenden, da dieser etwas kürzer ist — die Werte in $2D/$2E müssen dort nicht gesetzt werden, da es in diesem Fall ohnehin der falsche Wert wäre. Diese Probleme treten jedoch nur dann auf, wenn die einzufügenden Maschinenroutinen *hinter* dem Basic-Programm liegen, sonst sind nämlich die Werte für $2D/$2E bei Typ »B« beziehungsweise »C« richtig.

Hat man nun Typ »A« oder »B« gewählt, so erwartet der »64'er-Packer« noch zwei Eingaben:

1. Inhalt der Adresse $01 (Prozessor-I/O-Port)

Hiermit läßt sich die Speicherkonfiguration nach dem Dekomprimieren einstellen. Die Eingabe erfolgt entweder dezimal oder — durch ein vorangestelltes »$« — hexadezimal. Normalerweise wird der Wert 55 beziehungsweise $37 eingegeben. Dieser ist auch als Default-Wert vorgegeben und braucht nur noch mit <RETURN> bestätigt zu werden. Siehe dazu aber folgende Tabelle:

52	($34)	Komplette 64 KByte RAM
53	($35)	RAM bis auf den I/O-Bereich ($D000-$DFFF)
54	($36)	Basic ausgeblendet (dafür RAM), aber I/O-Bereich und Betriebssystem
55	($37)	Einschaltwert (z.B. nach einem Reset): Basic, I/O-Bereich und Betriebssystem eingeschaltet

Natürlich kann man dieselben Konfigurationen auch mit anderen Werten erhalten, da nur die letzten 3 Bit relevant sind. Zwei dieser Eingabewerte haben allerdings hier Sonderfunktionen:

00	($00)	wie $34, nur wird der Dekompressor um 4 Byte kürzer
255	($ff)	wie $37, nur wird der Dekompressor um 2 Byte kürzer

Gibt man ein <+> hinter diesem Wert ein (zum Beispiel $36+), so wird das Programm mit gesetztem Interrupt-Flag angesprungen und eventuell 1 Byte eingespart, im Falle eines nachgestellten <-> wird ein CLI (Clear Interrupt-Flag) erzwungen. Eine Option, die Sie wahrscheinlich selten verwenden werden.

Anderenfalls wird die Speicherkonfiguration automatisch überprüft: sind I/O-Bereich und Betriebssystem eingeschaltet, so wird ein CLI eingefügt, sonst wird das während des Dekompressionsvorganges gesetzte Interrupt-Flag beibehalten.

2. Einsprungadresse

Hier gibt man die Startadresse des (Maschinen-) Programms ein, und zwar entweder dezimal oder - durch ein vorangestelltes »$« - hexadezimal. Beispielsweise sind 2061 und $80D identische, gültige Eingaben. Voreinstellung ist $080d (2061).

Start des Kompreßvorgangs

Nach diesen Eingaben wird der eigentliche Kompressionsvorgang gestartet. Das Komprimieren erfolgt direkt auf Diskette. Währenddessen wird ein Teil der aktuell bearbeiteten Daten auf dem Bildschirm angezeigt (optische Kontrolle).

<Z> Zwischenkompressor (auch als Vorkompressor verwendbar)

Die Auswahl des zu komprimierenden Programmes geschieht wie oben beschrieben. Es ist hier jedoch nur ein File auszuwählen (im Falle einer Mehrfachauswahl wird das in der Directory am weitesten hinten stehende geladen). Die Eingabe des neuen Filenamens ist mit der beim Vorkompressor identisch. Bei der Typeingabe hat man nun allerdings sechs Möglichkeiten:

Typ »A«: Zwischenkompressor 1

Dies ist in der Regel der effektivste der drei im 64'er-Packer enthaltenen Zwischenkompressoren. Mit ihm kann man mit dem Vorkompressor oder mit einem der Zwischenkompressoren schon komprimierte Programme kürzen. Dabei wird vom Benutzer noch eine Eingabe verlangt - die sogenannte »Suchtiefe«, die zwischen 1 und 255 liegen kann. Eine kleine Suchtiefe hat einen sehr schnellen - bei einer Suchtiefe von 1 sogar linear von der Programmlänge abhängigen (ca. 1 Block pro Sekunde) - Kompressionsvorgang zur Folge. Je größer die Suchtiefe, um so effektiver arbeitet der Packer in der Regel. Jedoch wird die Kompressionsleistung nicht mehr erhöht, wenn die Suchtiefe größer wird als die Programmlänge des zu komprimierenden Programmes in Blöcken auf Diskette. Das heißt konkret, es macht keinen Unterschied, ob man ein 120-Block-Programm mit Tiefe 120 oder 255 komprimiert. Bei höchster Kompressionsleistung ist die Kompressionszeit - in gewissen Grenzen - quadratisch von der Programmlänge abhängig. Im Extremfall sind Kompressionszeiten von zirka 5 Stunden durchaus möglich.

Typ »B«: Zwischenkompressor 2

Wie Typ »A«, jedoch mit einem etwas einfacheren Verfahren und 27 Byte kürzerem Dekompressor. Dieser Typ ist unter Umständen Typ »A« vorzuziehen, wendet man ihn auf relativ kurze, vorkomprimierte Programme an.

Typ »C«: Zwischenkompressor 3

Schneller Kompressor mit einer Kompressionsgeschwindigkeit von zirka einem Block pro Sekunde. Auch er erwartet als weiteren Parameter bei der Eingabe eine Suchtiefe. Normalerweise sollte man hier Null eingeben. Je größer die eingegebene Zahl ist, um so unwahrscheinlicher ist es, daß der Packer viel bringt. Jedoch kann es durchaus sinnvoll sein, ein mit Zwischenkompressor 3/Tiefe 0 (oder ein mit Zwischenkompressor 1 oder 2) komprimiertes Programm mit einer Tiefe größer als 0 noch einmal zu komprimieren.

Typ »D«

Wie Typ »A«, jedoch nicht nur auf vor- beziehungsweise zwischenkomprimierte Programme anwendbar, sondern auf beliebige, einteilige, nach $07C0 beginnende Programme mit einer Länge von nicht mehr als $F7FF (63784 = 62 KByte) Bytes. Da der Packer somit eigentlich zu einem »Vorkompressor« wird, werden vom Benutzer auch die gleichen Eingaben wie beim Vorkompressor (siehe dort) verlangt.

Typ »E«

Wie Typ »B«, darüber hinaus gilt das bei Typ »D« Beschriebene.

Typ »F«

Wie Typ »C«, darüber hinaus gilt das bei Typ »D« Beschriebene.

Fügt man ein <+> zu dieser Eingabe hinzu (zum Beispiel »D+«), so arbeitet der Kompressor etwa 5% schneller, da der Bildschirm abgeschaltet wird. Allerdings verschwindet somit die optische Kontrolle - es wird sonst jederzeit die Länge des bereits komprimierten Programmstückes in Byte und Diskettenblöcken sowie das Ende der Kompression angezeigt.

Nach dem Laden des zu komprimierenden Files erfolgt die Aufforderung, die Zieldiskette einzulegen und <Q> zu drücken. Danach beginnt der Kompressionsvorgang, der in diesem Fall ausschließlich im RAM abläuft. Ist der Kom-



pressionsvorgang beendet, so erscheint ein »*« in der linken, oberen Bildschirmecke. Achten Sie darauf, daß die Floppy nun eingeschaltet und eine Diskette eingelegt ist. Drücken Sie dann die SPACE-Taste, so beginnt der Speichervorgang. Tritt dabei ein Fehler auf, erscheint ein »–« statt des Sterns. Mit einem weiteren Druck auf <SPACE> kann der Speicherversuch notfalls beliebig oft wiederholt werden.

Nachkompressor (nur für vor- oder zwischenkomprimierte Programme)

Auswahl und Eingabe des neuen Filenamens erfolgen wie beim Zwischenkompressor, alles andere geschieht automatisch. Nach dem Laden des zu komprimierenden Programmes wird dessen alte Länge (vermindert um ein paar jetzt unnötige Routinen des Vor- oder Zwischenkompressors) und die neue Länge in Byte angezeigt. Das Komprimieren erfolgt direkt auf Diskette.

Directory

Holt das Inhaltsverzeichnis der Diskette auf den Bildschirm. Danach kann man durch Drücken einer beliebigen Taste wieder ins Funktionsauswahlmenü gelangen.

Exit

Verlassen des 64'er-Packers, ein Basic-Warmstart wird ausgelöst. Bedenken Sie, daß wegen des eingeschalteten EPROMs nur 30719 Basic Bytes frei sind. Wollen Sie den vollen Basic-Speicherplatz, so müssen Sie zuerst das EPROM abschalten ($8000-$9FFF freigeben) und dann einen Reset auslösen (Druck auf den Resettaster, SYS 64738 oder Aus- und Einschalten des Rechners).

Tips für das optionale Packen

Abhängig davon, ob Sie einen schnellen Kompressionsvorgang, einen schnellen Dekompressionsvorgang oder optimale Effektivität des Packers bevorzugen, sind folgende Kompressionsvorgänge besonders sinnvoll:

1. Schneller Packer (vor allem bei längeren Programmen):
   a) Zwischenkompressor Typ D / Tiefe 1 oder 2 (darauf unter Umständen Nachkompressor anwenden)
   b) Zwischenkompressor Typ E / Tiefe 1 oder 2 (darauf unter Umständen Nachkompressor anwenden)
   c) Zwischenkompressor Typ F / Tiefe 0 (darauf unter Umständen Nachkompressor anwenden)

2. Schneller Entpacker:
   a) Vorkompressor
   b) Zwischenkompressor Typ D,E oder F
   c) Vorkompressor, darauf Zwischenkompressor Typ A, B oder C anwenden.

3. Effektivste Packer:
   a) Vorkompressor / Zwischenkompressor Typ A (oder B)/Nachkompressor
   b) Zwischenkompressor Typ F (Tiefe 0) / Zwischenkompressor Typ A (oder B) / Nachkompressor
   c) Zwischenkompressor Typ D / Nachkompressor
   d) Zwischenkompressor Typ D / Zwischenkompressor Typ C (Tiefe 0) / Nachkompressor

Welches Vorgehen beim Packen nun wirklich das effektivste ist – das zu packende Programm also am stärksten verkürzt – ist auch wesentlich vom zu packenden Programm selbst abhängig. Mit einer der vier letztgenannten Vorgehensweisen läßt sich in den meisten Fällen ein optimales Ergebnis erzielen. Aber auch hier gilt: im Zweifelsfalle verschiedene Reihenfolgen der einzelnen Verfahren ausprobieren.

Beim 64'er-Packer handelt es sich um ein komplexes Programmsystem. Auf eine »Schwäche« des 64'er-Packers sei hier noch hingewiesen: Es besteht die Möglichkeit, daß der Dekompressor des Vorkompressors nicht ordnungsgemäß funktionieren kann, wenn beispielsweise das zu komprimierende Programm bei $FFFF (oder kurz davor) aufhört und sich vor allem am Ende schlecht komprimieren läßt. Hierbei handelt es sich mit Sicherheit nicht um einen Programmierfehler, denn solche »Effekte« gibt es in jedem Packer an ähnlicher Stelle – sie sind unvermeidbar. Abhilfe läßt sich hier aber in den meisten Fällen schaffen, indem man statt des Vorkompressors gleich einen der Zwischenkompressoren (am sinnvollsten Typ D/möglichst große Suchtiefe) verwendet – falls das zu packende Programm, wie oben schon erwähnt, einteilig ist und nach $07C0 anfängt.

(Peter Arndt/Ulrich Völker/sk)

```
Name : 64'er-packer          8000 a000
--------------------------------------
8000 : 28 84 28 84 c3 c2 cd 38   ff
8008 : 30 a2 00 ad 75 dd 81 fb   83
8010 : ad 21 d0 9d 00 d8 9d 00   73
8018 : d9 9d 00 da 9d 00 db e8   36
8020 : d0 f1 a9 07 8d 18 d0 a2   56
8028 : 28 9d 27 d8 ca d0 fa 60   e4
8030 : 20 09 84 85 03 85 04 a9   66
8038 : 13 85 d3 20 cf ff c9 20   6b
8040 : f0 f9 c9 24 f0 53 c9 30   55
8048 : 90 56 c9 3a b0 53 29 0f   25
8050 : 85 02 a4 04 a5 03 0a 26   67
8058 : 04 0a 26 04 65 03 aa 98   b6
8060 : 65 04 85 04 8a 0a 26 04   43
8068 : 65 02 85 03 90 02 e6 04   4d
8070 : 20 cf ff 90 d1 c9 47 b0   74
8078 : 28 c9 3a 90 06 c9 41 90   fa
8080 : 20 e9 07 29 0f 85 02 a5   ec
8088 : 03 0a 26 04 0a 26 04 0a   90
8090 : 26 04 0a 26 04 65 02 85   7e
8098 : 03 20 cf ff c9 30 b0 d5   2c
80a0 : 60 68 38 e9 05 48 60 48   24
80a8 : 4a 4a 4a 4a 20 b2 80 68   5e
80b0 : 29 0f 09 30 c9 3a 90 02   5e
80b8 : e9 39 95 28 e8 60 a9 f6   ce
80c0 : 8d 20 d0 8d 21 d0 a0 00   5e
80c8 : b9 c4 82 f0 f0 c9 01 d0   a5
80d0 : 0c c8 be c4 82 a9 20 20   bf
80d8 : 16 e7 ca d0 fa 20 16 e7   87
80e0 : c8 d0 e5 0d d4 45 53 54   99
80e8 : 41 42 53 54 41 4e 44 20   82
80f0 : 20 20 20 20 3a 20 20 32   b6
80f8 : 30 30 00 ff ff ff ff 0d   5a
8100 : d4 41 42 45 4c 4c 45 4e   87
8108 : 4c 55 45 43 4b 45 20 20   58
8110 : 3a 20 20 32 30 30 30 00   ee
8118 : 0d d3 55 43 48 57 45 49   b3
8120 : 54 45 20 20 20 20 20 20   e7
8128 : 20 3a 20 20 39 00 0d c9   cd
8130 : 4e 48 41 4c 54 20 56 4f   ba
8138 : 4e 70 24 30 30 30 31 3a   63
8140 : 20 20 35 35 00 0d c5 49   76
8148 : 4e 53 50 52 55 4e 47 41   06
8150 : 44 52 45 53 53 45 3a 20   02
8158 : 20 24 30 38 30 44 00 93   ea
8160 : 0d 9f 20 12 20 20 20 20   4b
8168 : d3 45 51 55 45 4e 5a 4b   a4
8170 : 4f 4d 50 52 45 53 53 4f   9f
8178 : 52 2f 20 cc 49 4e 4b 45   c2
8180 : 52 20 d6 34 2e 31 20 20   4c
8188 : 20 20 00 93 0d 9e 20 12   95
8190 : 20 20 20 20 20 20 20 20   90
8198 : 20 20 c2 49 54 4b 4f 4d   1a
81a0 : 50 52 45 53 53 4f 52 20   0e
81a8 : d6 33 2e 32 20 20 20 20   ae
81b0 : 20 20 20 20 20 20 00 93   17
81b8 : 0d 05 20 12 20 20 20 20   56
81c0 : 20 20 d3 45 51 55 45 4e   ff
81c8 : 5a 2d c9 c9 2d cb 4f 4d   6d
81d0 : 50 52 45 53 53 4f 52 20   3e
81d8 : d6 34 2e 30 20 20 20 20   1e
81e0 : 20 20 00 0d 0d c4 49 53   75
81e8 : 4b 45 54 54 45 20 45 49   72
81f0 : 4e 4c 45 47 45 4e 20 55   91
81f8 : 4e 44 20 d4 41 53 54 45   96
8200 : 20 44 52 55 45 43 4b 45   a8
8208 : 4e 0d 00 0d d4 59 50 20   18
8210 : 41 2c 20 42 20 4f 44 45   d0
8218 : 52 20 43 20 3a 20 20 41   f7
8220 : 9d 00 0d d4 59 50 20 41   b6
8228 : 20 42 49 53 20 46 20 20   1b
8230 : 20 20 20 3a 20 20 41 9d   f3
8238 : 00 0d 0d ce 45 55 45 52   95
8240 : 20 c6 49 4c 45 4e 41 4d   06
8248 : 45 20 20 3a 20 20 00 0c   08
8250 : 0f 01 04 09 0e 07 3a 20   44
8258 : 00 20 20 20 5a 09 05 0c   8f
```

Listing. Der »64'er-Packer« ist eines der leistungsfähigsten Programme, die bisher auf diesem Sektor veröffentlicht wurden. Bitte mit dem MSE (Seite 159) eingeben.

```
8260 : 04 09 13 0b 20 05 09 0e   7a
8268 : 0c 05 07 05 0e 20 15 0e   ac
8270 : 04 20 27 11 27 20 04 12   18
8278 : 15 05 03 0b 05 0e 00 50   94
8280 : 12 0f 07 12 01 0d 0d 0c   e3
8288 : 01 05 0e 07 05 20 01 0c   de
8290 : 14 3a 20 24 00 20 20 20   10
8298 : 0e 05 15 3a 20 24 00 20   19
82a0 : 20 20 20 42 0c 0f 03 0b   7c
82a8 : 13 3a 20 20 30 30 31 20   6e
82b0 : 20 20 20 20 20 20 42 19   2b
82b8 : 14 05 13 3a 20 20 30 30   7f
82c0 : 30 30 2e 00 93 9a 0e 08   ea
82c8 : 0d 0d 20 12 20 20 20 20   6a
82d0 : 20 20 20 20 20 20 20 20   d0
82d8 : 20 36 34 27 45 52 20 d0   0e
82e0 : 41 43 4b 45 52 20 20 20   25
82e8 : 20 01 08 0d 0d 0d 01 03   70
82f0 : 12 05 76 92 9a 4f 52 4b   79
82f8 : 4f 4d 50 52 45 53 53 4f   27
8300 : 52 20 20 01 07 12 05 64   68
8308 : 92 9a 49 52 45 43 54 4f   e2
8310 : 52 59 0d 0d 01 03 12 05   6e
8318 : 7a 92 9a 57 49 53 43 48   3a
8320 : 45 4e 4b 4f 4d 50 52 45   74
8328 : 53 53 4f 52 20 20 20 20   07
8330 : 20 12 05 65 92 9a 58 49   39
8338 : 54 0d 0d 01 03 12 05 6e   28
8340 : 92 9a 41 43 48 4b 4f 4d   8f
8348 : 50 52 45 53 53 4f 52 20   b6
8350 : 20 20 20 20 20 20 0d 0d   de
8358 : 0d 0d 0d 01 0f 28 43 29   e1
8360 : 20 31 39 38 37 0d 0d 01   80
8368 : 12 42 59 0d 0d 01 04 20   bd
8370 : d0 45 54 45 52 20 c1 52   72
8378 : 4e 44 54 20 26 20 d5 4c   55
8380 : 52 49 43 48 20 d6 4f 45   d1
8388 : 4c 4b 45 52 0d 0d 0d 20   c3
8390 : 12 20 20 20 20 20 20 20   82
8398 : 20 20 20 20 cd 41 52 4b   9b
83a0 : 54 20 26 20 d4 45 43 48   a7
83a8 : 4e 49 4b 20 20 20 20 20   36
83b0 : 20 20 20 20 20 20 20 0d   8a
83b8 : 00 86 64 85 65 a2 05 d0   e6
83c0 : 13 85 65 84 64 f6 58 a5   2a
83c8 : 64 fd f4 83 a8 a5 65 fd   22
83d0 : f9 83 b0 ed a9 30 95 57   96
83d8 : 38 ca 10 eb a2 04 b5 58   c9
83e0 : c9 30 d0 05 ca d0 f7 b5   15
83e8 : 58 20 16 e7 ca 10 f8 a9   37
83f0 : 20 4c 16 e7 01 0a 64 e8   7d
83f8 : 10 00 00 00 03 27 e8 bd   91
8400 : e3 81 20 16 e7 d0 f7 60   14
8408 : e8 bd e3 80 20 16 e7 d0   cc
8410 : f7 60 a2 28 a9 20 95 27   25
8418 : ca d0 fb b9 4f 82 f0 f1   31
8420 : 95 28 c8 e8 d0 f5 ff ff   d5
8428 : 20 a3 fd 20 50 fd 20 15   3d
8430 : fd 20 5b ff a9 08 aa 20   da
8438 : ba ff 20 e7 ff 20 be 80   f4
8440 : a2 04 bc 6e 88 b9 cc 04   73
8448 : 49 80 99 cc 04 ca 10 f2   8e
8450 : a9 e2 85 a2 a5 a2 f0 e8   25
8458 : 20 e4 ff a2 7c c9 56 f0   90
8460 : 6e a2 d4 c9 5a f0 68 a2   a2
8468 : a8 c9 4e f0 62 c9 45 d0   d2
8470 : 03 6c 00 a0 49 44 d0 dc   71
8478 : 85 b9 20 44 e5 a9 01 a2   5f
8480 : 44 a0 e6 20 bd ff 20 c0   b0
8488 : ff b0 af a2 08 20 c6 ff   bc
8490 : 20 13 ee a5 90 d0 a3 20   08
8498 : 13 ee 20 13 ee 20 13 ee   a7
84a0 : a4 90 d0 1b a9 0d 20 16   d4
84a8 : e7 20 13 ee aa 20 13 ee   18
84b0 : 20 b9 83 20 13 ee f0 e2   c4
84b8 : 20 16 e7 a5 90 f0 f4 20   36
84c0 : cc ff a9 08 20 c3 ff 20   58
84c8 : e4 ff f0 fb 4c 3a 84 86   1d
84d0 : 92 20 5a 91 f0 f6 69 3f   26
84d8 : 85 bb a2 56 20 ff 83 aa   14
84e0 : 20 cf ff c9 0d f0 07 95   c0
84e8 : 17 e8 e0 10 d0 f2 8a f0   5e
84f0 : db 86 16 24 92 50 03 4c   69
84f8 : 85 88 30 2d a2 3f 20 ff   18
8500 : 83 20 cf ff c9 0d f0 c4   d9
8508 : c9 41 90 f0 c9 47 b0 ec   27
8510 : e9 43 85 05 10 02 69 03   5a
8518 : 85 07 20 cf ff 49 2b f0   fc
8520 : 02 a9 1b 85 09 a5 05 30   a1
8528 : 52 a2 28 20 ff 83 20 cf   16
8530 : ff c9 0d f0 97 c9 41 90   63
8538 : f0 c9 44 b0 ec a8 b9 2a   83
8540 : 88 85 05 f0 3e a2 4b 20   51
8548 : 30 80 aa a5 03 85 06 f0   6e
8550 : 18 c9 ff d0 04 e6 05 e6   c0
8558 : 05 e0 2d f0 13 e0 2b f0   fd
8560 : 06 29 06 49 06 f0 09 a9   05
8568 : 04 49 05 18 65 05 85 05   f4
8570 : a2 62 20 30 80 a6 03 a5   e6
8578 : 04 d0 04 a2 27 a9 08 86   27
8580 : 0a 85 0b 24 92 30 03 4c   e4
8588 : 80 88 20 09 80 a2 13 bd   5a
8590 : 39 88 95 4f ca d0 f8 20   b4
8598 : cb 90 20 50 00 a2 3e bd   47
85a0 : fb 87 95 4f ca d0 f8 20   05
85a8 : f5 90 a9 00 a8 99 e0 00   2b
85b0 : c8 d0 fa 38 a5 ae e9 e0   df
85b8 : a8 a5 af e9 00 85 cf a9   1b
85c0 : e0 85 ce a2 1d bd de 87   b5
85c8 : 95 4f ca d0 f8 4c 50 00   05
85d0 : a2 02 a0 ff 84 dd b9 e0   7b
85d8 : 00 88 f0 07 d9 e0 00 90   ff
85e0 : f8 b0 f1 a4 dd 94 a7 a9   b6
85e8 : ff 99 e0 00 ca 10 e3 a2   ee
85f0 : 1e 9a a9 03 85 07 4a a2   25
85f8 : 1f a0 00 20 57 94 a0 0a   1c
8600 : 20 12 84 2c 01 dc 70 fb   81
8608 : 20 4c 90 20 d9 f3 a9 08   6a
8610 : 20 79 f2 a5 08 8d a1 02   d6
8618 : a0 00 b1 e0 20 dd ed c8   7b
8620 : c4 e4 d0 f6 a2 1e bd 4c   14
8628 : 88 95 cf ca d0 f8 a6 c1   bb
8630 : 20 66 87 a5 ce c5 ae a5   3b
8638 : cf e5 af a0 00 b0 4d 20   f5
8640 : d0 00 85 0f c8 20 d0 00   24
8648 : 85 10 c8 20 d7 00 96 0f   01
8650 : e4 0f f0 58 c0 80 d0 f2   3c
8658 : a7 0f a0 01 c5 a9 f0 18   6d
8660 : c5 a8 f0 14 c5 a7 f0 10   b5
8668 : 20 dd ed 18 98 65 ce 85   f0
8670 : ce 90 02 e6 cf 4c 2e 86   09
8678 : a5 a8 20 dd ed e4 10 d0   1d
8680 : 01 c8 98 20 dd ed 8a 20   c7
8688 : dd ed d0 df 8c a1 02 4c   02
8690 : 3d 94 ca 38 a5 fd e5 02   b7
8698 : 85 fd b0 02 c6 fe a4 02   83
86a0 : e0 02 90 b0 d0 41 c0 02   da
86a8 : f0 aa d0 3b 84 02 18 98   73
86b0 : 65 ce 85 fd a5 cf 69 00   1c
86b8 : f0 9e 85 fe a6 c2 a5 ae   ad
86c0 : c5 fd a5 af e5 fe 90 ca   11
86c8 : a0 00 20 e7 00 d9 0f 00   78
86d0 : d0 cc c8 c4 02 d0 f3 18   78
86d8 : a5 fd 65 02 85 fd 90 04   a8
86e0 : e6 fe f0 b0 e8 d0 d7 a5   57
86e8 : fd 85 ce a5 fe 85 cf 86   78
86f0 : fe b6 0e e8 96 0f a0 00   4e
86f8 : a2 00 86 fd b5 0f e8 d5   1f
8700 : 0f d0 41 e8 d5 0f d0 3c   76
8708 : e8 d5 0f f0 fb c9 00 d0   6c
8710 : 40 a5 a9 99 03 00 c8 8a   29
8718 : 38 e5 fd 99 03 00 c8 e4   12
8720 : 02 d0 d7 a5 a7 20 dd ed   04
8728 : a5 fe 20 dd ed 84 fe 88   20
8730 : 98 20 dd ed a0 00 b9 03   04
8738 : 00 20 dd ed c8 c4 fe d0   ce
8740 : f5 4c 2e 86 c5 a9 f0 09   37
8748 : c5 a8 f0 05 a6 fd e8 d0   de
8750 : ca 48 a5 a8 99 03 00 c8   00
8758 : 8a 38 e5 fd 99 03 00 c8   7b
8760 : 68 d0 b8 e8 e6 cf a0 00   eb
8768 : 20 d0 00 c8 20 de 00 d0   a4
8770 : 40 c8 20 de 00 d0 3a c8   f9
8778 : f0 05 20 de 00 f0 f8 c9   ce
8780 : 00 d0 11 98 f0 dd 20 b1   22
8788 : 87 a5 a9 20 dd ed 98 20   40
8790 : dd ed d0 14 48 20 b1 87   76
8798 : a5 a8 20 dd ed 98 20 dd   35
87a0 : ed 68 20 dd ed 98 f0 bc   66
87a8 : 18 65 ce 85 ce 90 b7 b0   89
87b0 : b3 8a f0 11 e0 03 b0 0e   0c
87b8 : a5 a9 20 dd ed a9 00 20   62
87c0 : dd ed ca d0 f3 60 a5 a7   89
87c8 : 20 dd ed 8a 20 dd ed a9   9f
87d0 : 01 20 dd ed a5 a9 20 dd   fa
87d8 : ed a2 00 8a 4c dd ed e6   a1
87e0 : 01 98 f0 0d 88 b3 ce fe   6a
87e8 : e0 00 d0 f5 de e0 00 d0   52
87f0 : f0 c6 cf d0 ef c6 01 e6   58
87f8 : cf 4c d0 85 e6 01 a5 f7   cf
8800 : 85 b3 a5 f8 85 b4 a9 e0   4e
8808 : 85 ae b1 f7 91 ae e6 f7   6a
8810 : d0 02 e6 f8 e6 ae d0 02   e5
8818 : e6 af a5 f7 c5 f9 d0 ea   83
8820 : a5 f8 c5 fa d0 e4 88 c6   f6
8828 : af b1 ae 69 1e e6 af c8   f2
8830 : 91 ae c8 69 f7 91 ae c6   cc
8838 : 01 60 e6 01 a9 00 a0 e0   22
8840 : 99 00 01 c8 d0 fa e6 58   64
8848 : d0 f6 c6 01 60 e6 01 b1   0a
8850 : ce c6 01 60 e6 01 b3 ce   b1
8858 : c6 01 60 e6 01 d1 ce 08   7e
8860 : c6 01 28 60 e6 01 b1 fd   f6
8868 : c6 01 60 50 48 00 00 17   84
8870 : 50 67 a0 ff ff ff ff ff   9c
8878 : ff ff ff ff ff ff ff ff   77
8880 : a2 35 20 30 80 20 09 80   f9
8888 : 20 d3 90 20 ec 90 20 2c   e6
8890 : 91 a0 0a 20 12 84 2c 01   f0
8898 : dc 70 fb 24 92 70 03 4c   81
88a0 : 18 98 ad c9 09 85 5a a2   14
88a8 : a6 bd 73 8b 9d 4e 09 ca   81
88b0 : d0 f7 a9 00 9d 00 06 9d   13
88b8 : 00 07 9d 00 03 8a 9d 00   9e
88c0 : 08 e8 d0 ee 86 fb a4 ae   86
88c8 : 84 fd a5 af 85 fc 38 e9   9f
88d0 : 0a 85 fe a2 24 bd 4f 8b   35
88d8 : 9d 49 01 ca 10 f7 20 50   95
88e0 : 01 84 60 84 61 c8 84 5d   f5
88e8 : b9 00 07 99 00 05 be 00   ba
88f0 : 06 d0 08 29 ff d0 04 c6   aa
88f8 : 5d d0 0f e4 60 e5 61 b0   3a
8900 : 09 86 60 b9 00 07 85 61   ad
8908 : 84 64 8a 99 00 04 c8 d0   79
8910 : d7 88 84 5e a5 5d f0 05   2b
8918 : a6 64 de 00 03 ad 00 06   52
8920 : 85 60 ad 00 07 85 61 a0   a4
8928 : ff 84 62 84 63 c8 84 64   ea
8930 : c8 be 00 06 e4 60 b9 00   50
8938 : 07 e5 61 b0 17 a5 60 85   cb
8940 : 62 a5 61 85 63 a5 64 85   7e
8948 : 65 84 64 86 60 b9 00 07   bb
8950 : 85 61 90 12 e4 62 b9 00   34
8958 : 07 e5 63 b0 09 84 65 86   98
8960 : 62 b9 00 07 85 63 c8 d0   b8
8968 : c8 a5 64 aa fe 00 03 86   7a
8970 : 5f bd 00 08 c5 5f d0 f3   31
8978 : a5 65 9d 00 08 aa fe 00   09
8980 : 03 86 5f bd 00 08 c5 5f   6c
8988 : d0 f3 a4 64 a6 65 18 b9   71
8990 : 00 06 7d 00 06 99 00 06   2c
8998 : b9 00 07 7d 00 07 99 00   61
89a0 : 07 a9 ff 9d 00 07 c6 5e   3f
89a8 : f0 03 4c 1d 89 a9 80 85   c4
89b0 : 61 a0 00 84 5e 84 6f 84   c3
89b8 : ea 84 d9 84 60 a2 01 b5   76
89c0 : 6e 95 6f b5 d8 95 d9 b5   98
89c8 : e9 95 ea 8a d9 00 03 d0   d3
89d0 : 24 a5 5e f5 6e 99 00 06   dd
89d8 : 98 a4 5e 99 00 07 c8 c4   72
89e0 : 5d f0 1c 84 5e a8 18 b5   44
89e8 : d9 65 60 95 d9 b5 ea 65   00
89f0 : 61 95 ea f6 6f c8 d0 d3   dd
89f8 : 46 61 66 60 e8 d0 c0 86   ba
8a00 : 5b ca 8e a1 09 a9 00 a8   c7
8a08 : e5 fd 8d df 09 98 e5 fe   36
8a10 : 8d e3 09 a5 5a e9 10 6d   96
8a18 : df 09 8d f0 09 98 6d e3   d0
8a20 : 09 8d f1 09 a9 01 e5 5a   7c
8a28 : 8d e9 09 a9 b0 65 5b 8d   e0
8a30 : bc 09 aa e8 8e a5 09 65   3d
8a38 : 5b 8d c3 09 aa e8 8e aa   ed
8a40 : 09 65 5b 8d d8 09 84 66   39
8a48 : 84 67 84 68 be 00 03 a5   f1
8a50 : 66 79 00 04 85 66 a5 67   e4
8a58 : 79 00 05 85 67 90 03 e6   98
8a60 : 68 18 ca d0 ea c8 d0 e4   a3
8a68 : a7 66 46 68 66 67 6a 46   b9
8a70 : 68 66 67 6a 46 68 66 67   43
8a78 : 6a 85 66 ca 8a 25 b0 aa   82
8a80 : 98 38 2a ca 10 fc 85 5c   d0
8a88 : 49 80 0a ad d8 09 e9 01   c9
8a90 : 65 66 90 03 e6 67 18 a6   04
8a98 : 5d f0 04 65 5d 90 02 e6   4b
8aa0 : 67 85 66 a0 30 20 12 84   cd
8aa8 : a5 fe 20 a7 80 a5 fd 20   37
8ab0 : a7 80 a0 46 20 1b 84 a5   c0
8ab8 : 67 20 a7 80 a5 66 20 a7   87
8ac0 : 80 20 d9 f3 a9 08 20 79   93
8ac8 : f2 a2 4f bd 00 09 20 dd   1b
8ad0 : ed e8 d0 f7 b5 6f 20 dd   77
8ad8 : ed e8 e4 5b d0 f6 a2 00   2d
8ae0 : b5 d9 20 dd ed e8 e4 5b   b6
8ae8 : d0 f6 a2 00 b5 ea 20 dd   cb
8af0 : ed e8 e4 5b d0 f6 a2 00   45
8af8 : bd 00 07 20 dd ed e8 e4   36
8b00 : 5d d0 f5 a6 b1 8a 49 0f   ca
8b08 : a8 56 ea 76 d9 88 d0 f9   7e
8b10 : ca d0 f2 e6 fc a0 00 20   f1
8b18 : 49 01 a8 e6 fb d0 02 e6   05
8b20 : fc be 00 03 b9 00 06 75   7a
8b28 : d8 85 5f b5 e9 4a 66 5f   9b
8b30 : 66 5e 46 5c 90 09 66 5c   86
8b38 : a8 a5 5e 20 dd ed 98 ca   94
8b40 : d0 eb a5 fb c5 ae a5 fc   51
8b48 : e5 af 90 c9 4c 3d 94 e6   31
8b50 : 01 b1 fb c6 01 60 e6 01   b2
8b58 : 88 c0 ff f0 0c b3 fb fe   ab
8b60 : 00 06 d0 f4 fe 00 07 d0   e4
8b68 : ef c6 fc a5 fc c9 09 d0   92
8b70 : ec c6 01 60 01 08 0c 08   9c
8b78 : c3 07 9e 32 30 36 32 ff   2a
8b80 : 00 00 00 78 a0 01 84 a4   fd
8b88 : 88 84 ac 84 ad 84 01 a5   5c
```

```
8b90 : ac d0 02 c6 ad c6 ac a2  07
8b98 : 01 a9 ff 85 a0 85 a1 a9  2e
8ba0 : 7f 85 a2 46 a4 90 14 66  3f
8ba8 : a4 84 a3 a5 ae d0 02 c6  33
8bb0 : af c6 ae a0 00 b1 ae a4  14
8bb8 : a3 85 a3 06 a3 b0 06 a5  eb
8bc0 : a2 35 a0 95 a0 c0 ff f0  c9
8bc8 : 15 b9 c1 08 c5 a0 b9 d1  17
8bd0 : 08 e5 a1 b0 09 c8 38 66  ce
8bd8 : a2 b0 c8 ca f0 c1 a5 a0  53
8be0 : f9 c0 08 85 a0 a5 a1 f9  9e
8be8 : d0 08 be b0 08 c0 0f f0  27
8bf0 : 06 4a 66 a0 c8 d0 f6 18  e8
8bf8 : 8a 65 a0 aa bd e0 08 a0  f6
8c00 : 00 91 ac a9 ff c5 ac a9  5d
8c08 : ff e5 ad 90 82 a2 ff b1  18
8c10 : ac 9d c4 00 20 ff ff e8  90
8c18 : d0 f5 ff ff ff ff ff ff  e3
8c20 : 65 03 84 5f b0 3f 85 fd  2b
8c28 : 69 01 85 5f 90 04 a9 63  f6
8c30 : 85 5e b1 5e 85 60 84 62  4f
8c38 : a0 00 ad ff ff 91 5e c8  db
8c40 : d1 fc f0 07 c8 d0 f9 e6  75
8c48 : fd d0 f5 98 18 65 fc 85  e9
8c50 : fc 90 02 e6 fd c5 fe a5  47
8c58 : fd e5 ff c5 03 d0 4a a5  2c
8c60 : fc 25 fd c9 ff b0 42 a0  77
8c68 : 00 a5 60 91 5e c8 b1 fe  76
8c70 : d1 fc d0 05 c8 d0 f7 a0  c8
8c78 : 3f a6 fd e8 d0 0b a5 fc  9d
8c80 : 49 ff 85 57 c4 57 90 01  60
8c88 : a8 c0 40 90 02 a0 3f c4  5e
8c90 : 58 90 a5 f0 a3 a5 fc e5  df
8c98 : fe c0 03 b0 04 c9 40 b0  be
8ca0 : 97 84 58 85 61 c0 3f d0  fb
8ca8 : 8f a5 61 a4 58 f0 4e 88  4e
8cb0 : f0 4b 88 d0 28 a6 5d e0  71
8cb8 : 3f b0 42 09 40 a2 02 48  b3
8cc0 : a5 5d f0 0e 84 5d c9 3f  eb
8cc8 : 90 05 20 73 04 a9 3f 20  1c
8cd0 : 73 04 68 c4 62 f0 33 20  b3
8cd8 : 73 04 68 d0 2d 88 d0 10  fc
8ce0 : c9 40 b0 0c 09 80 a2 03  9c
8ce8 : d0 d5 a6 5d e0 3f b0 0d  dd
8cf0 : a8 e6 62 a7 58 09 c0 48  fa
8cf8 : 98 a0 00 f0 c2 ad ff ff  98
8d00 : 20 73 04 a2 01 e6 5d d0  8e
8d08 : 04 98 20 73 04 8a 18 65  8f
8d10 : fe 85 fe 90 02 e6 ff 25  44
8d18 : ff c9 ff b0 0b a5 5f f0  4f
8d20 : 04 a5 60 91 5e 4c 9a 05  fe
8d28 : a5 5d c9 3f 90 05 20 73  6f
8d30 : 04 a9 3f 20 73 04 4c 50  06
8d38 : 05 ff ff ff ff ff ff ff  3d
8d40 : 85 fd 65 03 90 05 a9 63  1c
8d48 : 85 5e 98 85 5f b1 5e 85  db
8d50 : 60 84 67 a0 00 ad ff ff  4e
8d58 : 91 5e c8 d1 fc f0 07 c8  8a
8d60 : d0 f9 e6 fd d0 f5 98 18  f5
8d68 : 65 fc 85 fc 90 02 e6 fd  fd
8d70 : 38 e5 fe 85 65 a7 fd e5  62
8d78 : ff c5 03 b0 79 85 66 8a  a3
8d80 : 25 fc c9 ff f0 70 a0 00  ab
8d88 : a5 60 91 5e c8 b1 fe d1  47
8d90 : fc d0 05 c8 d0 f7 a0 3f  1c
8d98 : a6 fd e8 d0 0b a5 fc 49  f5
8da0 : ff 85 57 c4 57 90 01 a8  20
8da8 : c0 40 90 02 a0 3f a2 01  7d
8db0 : c0 02 90 9f d0 0e a5 65  68
8db8 : 29 c0 d0 97 a5 66 05 5d  c5
8dc0 : d0 91 f0 54 c0 04 b0 0c  26
8dc8 : a5 66 d0 66 a5 5d 29 c0  0d
8dd0 : d0 60 f0 44 d0 22 a5 66  16
8dd8 : d0 12 a9 fe c5 fe a9 ff  f6
8de0 : e5 ff c5 65 90 07 a5 65  85
8de8 : c5 03 b0 01 ca a5 5d c9  5e
8df0 : 3f b0 25 e8 d0 22 b0 3d  49
8df8 : 98 e9 03 aa a5 66 d0 12  90
8e00 : a9 fe c5 fe a9 ff e5 ff  ac
8e08 : c5 65 90 07 a5 65 c5 03  27
8e10 : b0 01 ca c0 40 90 01 ca  2e
8e18 : e4 67 90 16 d0 04 c4 58  88
8e20 : b0 10 86 67 84 58 a5 65  d3
8e28 : 85 59 a5 66 85 5a e0 3c  b7
8e30 : f0 03 4c bd 05 a4 58 f0  25
8e38 : 65 88 88 d0 0b a5 59 a2  a6
8e40 : 02 84 5d 20 73 04 d0 66  47
8e48 : 88 d0 12 a7 5d c9 40 b0  38
8e50 : 4d a5 59 20 73 04 8a 09  5e
8e58 : 40 a2 03 d0 e4 a5 5a d0  4b
8e60 : 12 a9 fe c5 fe a9 ff e5  c8
8e68 : ff c5 59 a5 59 90 0b c5  27
8e70 : 03 b0 07 a5 59 20 73 04  ae
8e78 : a5 5a 20 73 04 88 d0 11  ab
8e80 : a5 5d c9 3f 90 05 20 73  c7
8e88 : 04 a9 3f 09 80 a2 04 d0  21
8e90 : b0 a5 5d 20 73 04 a0 00  48
8e98 : a7 58 09 c0 d0 a3 ad ff  a7
8ea0 : ff 20 73 04 a2 01 e6 5d  95
8ea8 : d0 04 98 20 73 04 8a 18  56
8eb0 : 65 fe 85 fe 90 02 e6 ff  8a
8eb8 : 25 ff c9 ff b0 07 a5 60  ea
8ec0 : 91 5e 4c 9a 05 a5 5d 8d  f5
8ec8 : 8e 03 4c 50 05 ff ff ff  45
8ed0 : ff ff ff ff ff ff ff ff  cf
8ed8 : ff ff ff ff ff ff ff ff  d7
8ee0 : ff ff ff ff ff ff ff ff  df
8ee8 : ff ff ff ff ff ff ff ff  e7
8ef0 : ff ff ff ff ff ff ff ff  ef
8ef8 : ff ff ff ff ff ff e2 fc  7c
8f00 : e6 31 4c d1 f4 20 5b 03  90
8f08 : a5 00 4c 0a e6 a5 f9 0a  99
8f10 : aa b5 06 85 06 b5 07 a2  37
8f18 : 00 85 07 a9 e0 85 00 8d  27
8f20 : 5b 02 8d 98 02 58 20 99  89
8f28 : d5 78 c9 02 b0 d7 ad 00  6c
8f30 : 04 f0 15 85 06 20 59 03  6f
8f38 : a0 02 b9 00 04 20 59 03  f4
8f40 : c8 d0 f7 ad 01 04 10 d1  38
8f48 : a0 ff c8 b9 00 04 20 59  a4
8f50 : 03 cc 01 04 d0 f4 4c 94  89
8f58 : c1 49 ff 85 1b a9 08 8d  a9
8f60 : 00 18 a9 01 2c 00 18 f0  fc
8f68 : fb a2 04 a9 00 46 1b 2a  de
8f70 : 0a 46 1b 2a 0a 8d 00 18  e7
8f78 : ca d0 f0 ea 8e 00 18 60  4e
8f80 : a2 00 86 b9 20 4c 90 20  e2
8f88 : d9 f3 20 21 90 8e 11 d0  ea
8f90 : 20 b7 90 a9 57 20 dd ed  af
8f98 : 8a 20 dd ed a9 03 20 dd  56
8fa0 : ed a9 20 a8 20 dd ed bd  a3
8fa8 : 00 8f 20 dd ed e8 88 d0  1d
8fb0 : f6 20 fe ed 8a 10 d9 20  04
8fb8 : b7 90 a9 45 20 dd ed a9  c6
8fc0 : 0d 20 dd ed a9 03 20 dd  01
8fc8 : ed 20 fe ed a5 08 8d a1  57
8fd0 : 02 20 92 90 30 48 48 d0  a1
8fd8 : 08 20 92 90 38 e9 03 85  91
8fe0 : b6 20 92 90 a8 20 92 90  54
8fe8 : 84 ac 85 ad 24 92 50 04  f9
8ff0 : a0 ba a9 09 84 ae 85 af  ac
8ff8 : 68 f0 63 a0 04 84 b6 38  75
9000 : a5 ae e5 b6 85 ae b0 02  e1
9008 : c6 af 20 92 90 20 e0 00  8e
9010 : c8 d0 f7 e6 af a0 02 20  63
9018 : 92 90 f0 3b 10 df 20 80  17
9020 : 90 a9 08 20 09 ed a9 6f  10
9028 : 20 c7 ed 20 13 ee c9 30  db
9030 : f0 19 a2 00 86 90 95 28  e9
9038 : e8 20 13 ee 24 90 50 f6  c9
9040 : a5 08 8d a1 02 68 68 20  c6
9048 : ef ed 38 60 ad a1 02 85  43
9050 : 08 a9 00 8d a1 02 60 20  cb
9058 : 92 90 aa ca 86 b6 a0 00  d7
9060 : 20 92 90 20 e0 00 c8 c4  ac
9068 : b6 d0 f5 98 18 65 ae 85  89
9070 : ae 90 02 e6 af 20 80 90  e3
9078 : a5 08 8d a1 02 4c a9 f5  ce
9080 : a9 1b 8d 11 d0 20 4c 90  9d
9088 : 4c 46 f6 e6 01 91 ae c6  77
9090 : 01 60 2c 00 dd 70 fb a9  71
9098 : 67 8d 00 dd a2 06 e8 8e  9d
90a0 : 00 dd ca d0 fd a2 04 ad  bc
90a8 : 00 dd 0a 2a 66 a6 4a 66  f0
90b0 : a6 ca d0 f3 a5 a6 60 a9  d2
90b8 : 08 20 0c ed a9 6f 20 b9  9b
90c0 : ed a9 4d 20 dd ed a9 2d  27
90c8 : 4c dd ed a9 ff 85 f8 a9  17
90d0 : 01 85 fa a0 06 b9 8b 90  e4
90d8 : 99 e0 00 88 10 f7 60 ae  92
90e0 : 01 dc 10 55 e8 f0 f8 20  38
90e8 : 49 91 d0 03 20 3c 91 20  f9
90f0 : 80 8f b0 eb 60 20 ec 90  bd
90f8 : a5 ac c5 f7 a5 ad e5 f8  b5
9100 : b0 08 a5 ac 85 f7 a5 ad  bd
9108 : 85 f8 a5 fa f0 0e 8a c5  07
9110 : f9 98 f0 04 e5 fa 90 04  92
9118 : 86 f9 84 fa a5 bb c9 40  fb
9120 : d0 d3 20 2c 91 a9 01 85  dd
9128 : af 4c 50 00 a9 61 85 b9  41
9130 : a5 16 a0 00 a2 17 4c f9  10
9138 : fd 4c 3a 84 a7 bb ca bc  77
9140 : 00 01 84 b7 18 e5 b7 85  73
9148 : bb a0 00 20 12 84 a4 b7  9f
9150 : 88 b1 bb 99 31 00 88 10  28
9158 : f8 60 20 09 84 20 b0 9f  f5
9160 : a9 04 a2 38 a0 93 20 3c  5b
9168 : 93 a2 00 86 c6 20 ff 83  92
9170 : 20 e4 ff f0 fb a9 00 85  38
9178 : b9 a2 e2 a0 0f 20 d5 ff  98
9180 : b0 b7 20 44 e5 38 a5 ae  b1
9188 : e9 20 85 ae 85 b4 a5 af  ac
9190 : e9 00 85 b5 06 ae 2a 06  1c
9198 : ae 2a 06 ae 2a e9 7f f0  85
91a0 : 77 85 b3 a0 e6 b9 00 0f  35
91a8 : 20 16 e7 c8 d0 f7 20 ec  0d
91b0 : 92 20 cd 92 20 75 92 20  50
91b8 : e4 ff 20 1b 92 e0 5f d0  57
91c0 : f6 a2 01 86 bc ca 86 b4  be
91c8 : 86 bb a9 10 85 b5 a4 03  37
91d0 : b1 b4 4a b0 2f a0 19 88  f1
91d8 : b1 b4 c9 22 d0 f9 84 ac  e2
91e0 : a0 03 c8 b1 b4 c9 22 d0  2e
91e8 : f9 c8 84 ad e0 9e b0 22  26
91f0 : b1 b4 9d 40 01 e8 c8 c4  6f
91f8 : ac d0 f1 98 e5 ad 9d 40  5e
9200 : 01 e8 86 bb a5 b4 69 1f  72
9208 : 85 b4 90 02 e6 b5 c6 b3  ea
9210 : d0 bc 20 44 e5 38 a5 bb  fd
9218 : 4c d4 84 c9 11 d0 25 a6  a2
9220 : b6 e8 e4 b3 b0 34 86 b6  2e
9228 : a2 04 a0 78 a9 a0 20 f1  07
9230 : 92 a5 b6 69 14 c5 b3 a2  f3
9238 : 17 90 03 4c ff e9 20 26  fe
9240 : 93 4c ca 92 c9 42 d0 13  16
9248 : a6 b3 ca 86 b6 20 c5 92  f4
9250 : a2 03 e8 20 ff e9 e0 17  b3
9258 : d0 f8 60 c9 91 d0 12 a6  2b
9260 : b6 f0 f7 c6 b6 a2 07 a0  43
9268 : 98 a9 70 20 f1 92 4c c5  65
9270 : 92 c9 54 d0 1e a9 00 85  50
9278 : b6 85 ac 20 c5 92 e6 ac  06
9280 : a5 ac c5 b3 f0 0c 20 26  9f
9288 : 93 20 cd 92 a5 d6 c9 18  59
9290 : d0 ec 60 aa 20 24 93 e0  77
9298 : 43 f0 07 e0 41 d0 1b a2  7e
92a0 : 24 2c a2 01 a5 b6 48 c8  66
92a8 : 8a 91 b4 e6 b6 20 24 93  29
92b0 : a5 b6 c5 b3 d0 f1 68 4c  6f
92b8 : 77 92 e0 20 d0 9c c8 b1  2d
92c0 : b4 49 25 91 b4 20 24 93  98
92c8 : a2 03 20 0c e5 a0 01 b1  40
92d0 : b4 4a 20 16 e7 c8 b3 b4  71
92d8 : c8 b1 b4 20 b9 83 a0 03  ea
92e0 : c8 b1 b4 20 16 e7 a5 d3  91
92e8 : c9 1b d0 f4 a9 0d 4c 16  72
92f0 : e7 86 d2 86 b5 85 d1 84  78
92f8 : b4 a2 14 a0 1d b1 d1 91  e0
9300 : b4 88 10 f9 a5 d2 85 b5  ae
9308 : a5 d1 85 b4 90 09 e9 28  d7
9310 : b0 0c c6 d2 38 b0 07 69  ca
9318 : 28 90 03 e6 d2 18 85 d1  ce
9320 : ca d0 d8 60 a5 b6 a0 00  27
9328 : 84 b4 4a 66 b4 4a 66 b4  06
9330 : 4a 66 b4 69 10 85 b5 60  cc
9338 : 24 2a 3d 50 46 73 90 03  13
9340 : 4c 60 b1 4c bd ff ff ff  8e
9348 : a0 00 b3 fc b1 fe 91 fc  c8
9350 : 8a 91 fe e6 fc d0 02 e6  6b
9358 : fd a6 fe d0 02 c6 ff ca  6e
9360 : 86 fe e4 fc a5 ff e5 fd  2c
9368 : b0 e0 60 8d 00 08 ee 74  37
9370 : 04 d0 03 ee 75 04 a5 4b  20
9378 : c9 39 f0 04 e6 4b d0 30  07
9380 : a9 30 85 4b a5 4a c9 39  52
9388 : f0 04 e6 4a d0 22 a9 30  a2
9390 : 85 4a a5 49 c9 39 f0 04  ff
9398 : e6 49 d0 14 a9 30 85 49  9e
93a0 : a5 48 c9 39 f0 04 e6 48  5e
93a8 : d0 06 a9 30 85 48 e6 47  b0
93b0 : e6 04 f0 01 60 e6 04 e6  10
93b8 : 04 a5 37 c9 39 f0 03 e6  8b
93c0 : 37 60 a9 30 85 37 a5 36  ad
93c8 : c9 39 f0 03 e6 36 60 a9  bf
93d0 : 30 85 36 e6 35 60 20 d9  b8
93d8 : f3 60 c6 01 a9 1b 8d 11  99
93e0 : d0 a9 2a 2c a9 2d 85 28  ff
93e8 : ae 01 dc e8 f0 fa a9 20  39
93f0 : 85 28 20 de 04 b0 ed a9  3e
93f8 : 08 20 79 f2 a5 e5 85 e6  3a
9400 : a9 40 85 ac a9 03 85 ad  e4
9408 : a0 00 b1 ac 20 dd ed c8  e4
9410 : d0 04 e6 ad c6 e6 c4 e4  d2
9418 : d0 f0 a5 e6 d0 ec a5 fc  ab
9420 : 85 ac a5 fd 85 ad a0 00  6d
9428 : e6 01 b1 ac c6 01 20 dd  41
9430 : ed a5 90 d0 af e6 ac d0  b4
9438 : ef e6 ad d0 eb 20 33 f3  94
9440 : 20 46 f6 4c e2 fc ff ff  e0
9448 : a0 00 b9 48 93 99 50 04  af
9450 : c8 d0 f7 a2 40 a9 03 86  3d
9458 : e0 85 e1 84 e5 a9 02 85  c2
9460 : 04 a2 b8 bd d3 96 20 8e  2b
9468 : 95 e8 d0 f7 18 20 a1 95  d9
9470 : a5 05 f0 51 10 02 a9 55  60
9478 : c9 50 85 0e b0 0d a2 f8  7c
9480 : bd db 96 20 8e 95 e8 d0  af
9488 : f7 a5 0e 29 07 c9 05 f0  af
9490 : 23 4a 08 d0 0d a9 a9 20  f9
```

Listing. (Fortsetzung)



Ergänzen Sie jetzt Ihre 64'er-Sammlung

Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Kennen Sie alle Ausgaben von 64'er? Suchen Sie einen ganz bestimmten Testbericht? Oder haben Sie einen Teil eines interessanten Kurses versäumt? Suchen Sie nach einer speziellen Anwendung?

Damit Sie jetzt fehlende Hefte mit »Ihrem« Artikel nachbestellen können, finden Sie auf diesen Seiten eine Zusammenstellung aller wesentlichen Artikel der Ausgaben 01 bis 12/85.

Und so kommen Sie schnell an die noch lieferbaren Ausgaben: Prüfen Sie, welche Ausgabe in Ihrer Sammlung noch fehlt, oder welches Thema Sie interessiert. Tragen Sie die Nummer dieser Ausgabe und das Erscheinungsjahr (z.B. 2/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird nach Zahlungseingang umgehend zur Auslieferung gebracht.

Aktuell

Stichwort	Titel	Seite	Ausgabe
Allgemeines	Commodore Gestern Heute Morgen	10	01/85
Computer	Amiga — Der neue Supercomputer	8	09/85
Interview	Interview mit David Crane (Game Designer)	146	06/85
Lernen	Schule braucht Computer (VAM-Computer)	9	06/85
Messen	International Chaos Communication Congress	15	03/85
	Heiße Messe in der Wüste: CES	8	03/85
	Hannover-Messe '85	8	06/85
	Hannover-Messe '85	8	07/85
	Chicago im Zeichen der CES	8	08/85
	Aktuelles von der C'85 in Köln	15	08/85
	Btx Total (Internationale Funkausstellung)	8	10/85
	PCW-Computermesse in London	8	11/85
	Neues von der Commodore-Fachausstellung 1985	8	12/85
Recht	Die neue Abmahnmasche — Vorsicht bei Programmangeboten	8	05/85
	Die Ex-Knacker — wo sind sie geblieben?	27	08/85
	Interview mit Raubkopierern (Section 8)	28	08/85
	Schützer kontra Knacki's	23	08/85
	Raub-Talkshow	12	08/85
	Das Urheberrechtsgesetz und Gedanken zu seiner Anwendung	21	08/85
	Änderung des Urheberrechtsgesetzes	162	09/85

Buchbesprechungen

Stichwort	Titel	Seite	Ausgabe
Anfänger	Goldmann Computer Compact	87	03/85
	Basic-Wegweiser für den C 64	86	05/85
	Alles über den C 64, Sachbuchreihe, Band 1	115	06/85
	Lehrspielzeug Computer: C 64/VC 20	112	11/85
	C 64 Computerhandbuch	171	11/85
	Einführungskurs: Commodore 64	144	12/85
Anwendung	Dienstprogramme VC 20, C 64 und SX	86	05/85
	Spaß an Mathe mit dem Commodore 64	88	07/85
	Mathe für die Oberstufe mit dem C 64	88	07/85
	Mathematische Routinen VC 20, Elektrotechnik/Elektronik	112	11/85
	Commodore 64-Listings, Band 2: Dateiverwaltung, Schule, Hobby	112	11/85
	Das Trainingsbuch zum Datamat	144	12/85
C 128	Bücher zum C 128	22	10/85
DFÜ	Das Mailbox-Jahrbuch: Nutz die Netze	112	11/85
Grafik	Grafik auf dem Commodore 64 (+ Fehlert. 9/85)	86	05/85
	Einführung in CAD mit dem Commodore 64	128	06/85
	Grafik & Musik auf dem Commodore 64	88	07/85
	Verschiedene Grafikbücher zum C 64	115	08/85
Programmieren	Von Basic zu Assembler: Das Commodore-Buch, Band 4	115	06/85
	64 Intern	115	06/85
	Das Interface Age System-Handbuch zum C 64	115	06/85
	Das C 64 Buch, Band 5: Simons Basic Leitfaden	144	12/85
	Basicode	144	12/85
	Noch mehr Tips und Tricks zum 64er	144	12/85
Speichern	Das Kassettenbuch zum C 64 und VC 20	87	03/85
	Die Floppy 1541 (M&T)	88	07/85
Spiele	Rombachs C 64 Spielführer	87	03/85
	Commodore 64-Listings, Band 1, Spiele	112	11/85
	35 ausgesuchte Spiele für Ihren Commodore 64	171	1/85

64'er Extra

Stichwort	Titel	Seite	Ausgabe
Prozessor	Befehlssatz des 6502/6510-Prozessors	84	09/85
Grafik	Die Videochip-Register des C 64	92	10/85
Sound	Der SID-Chip, seine Register und Programmierung	92	11/85
Speicher	Die Speicherbelegung des C 64	96	12/85

Abenteuerlösungen

Stichwort	Titel	Seite	Ausgabe
Lösungen	Dallas-Quest Lösung	90	01/85
	Guncho Krill-Enchanter ist gelöst	44	03/85
	Infocom-Geheimnisse gelüftet?	49	05/85
	Des Rätsels Lösung: Amazon	145	06/85
	Activision-Adventures entschleiert (Mindshadow, Tracer Sanction)	36	12/85
	Eureka! — ich hab's!	37	12/85
	Lösungen zu Hitchhiker's Guide und Sorcerer	39	12/85

Spiele-Tests

Stichwort	Titel	Seite	Ausgabe
007	James Bond — A View to a Kill	156	09/85
Abenteuer	Abenteuerpaket 1	48	08/85
	Shadowfire	146	09/85
	The Quest — mit C 64 auf Suche nach Drachen	47	01/85
Action	Hexenküche	50	07/85
	Master of the Lamps	48	07/85
	Rescue on Fractalus	158	10/85
	Stellar 7	49	08/85
Construction Set	Mail Order Monsters	49	08/85
	Racing Destruction Set	50	08/85
Geschicklichkeit	Australopedicus Robustus	50	08/85
	Boulder Dash II	159	10/85
	Crystal Castles	50	07/85
	Gribbly's Day out	148	09/85
	Rock'n Bolt	48	08/85
	Thing on a Spring	159	10/85
	Tom + Zaga	48	01/85
Pseudo-Adventures	Roland's Rat Race	49	08/85
	Fourth Protocol und Frankie g.t.H.	162	11/85
Renner	Die Renner 1985: Meistverkaufte Spiele	34	12/85
Schach	Viermal Schachmatt: Verschiedene Schachprogramme	32	12/85
Simulation	Elite	148	09/85
	Jump Jet	148	09/85
	Super Huey Hubschraubersimulator	49	07/85
Sport	Boxspiele: Frank Bruno's B. + Barry McGuigan Champions. B.	49	12/85
	Handkantenschlag per Joystick: Karateka + Exploding Fist	165	11/85
	Nick Faldo Plays the Open (Golf)	159	10/85
	Rallye Speedway	49	07/85
	Slapshot (Eishockey)	50	07/85
	Summer Games II	146	09/85
	World Series Baseball	49	07/85
Diverses	New York City und Air Support	145	06/85

Hardware-Tips und Bauanleitungen

Stichwort	Titel	Seite	Ausgabe
Audio/Video	Mit 5 Mark zu neuen Dimensionen (Stereoanlage am C 64)	34	05/85
	Ein Monitor ist genug (RGB + Composite an C 128)	16	10/85
C 16	Alte Datasette am C 16	31	04/85
	Alter Joystick am C 16	35	05/85
Eingabegeräte	Der Hexer — Zusatztastatur für den MSE	48	10/85
EPROM	EPROMs im Expansion-Port	46	10/85
	EPROM-Trans — Die Super-Erweiterung	42	10/85
	Das 64'er EPROM-Programmiergerät, Teil 1	44	12/85
Floppy/Datasette	Diskettenlaufwerk 1541 selbst justiert	32	10/85
	Die Datasette streikt nie wieder (Anpassung des Tonkopfs)	34	10/85
IEC-Bus	Auf zu neuen Welten: IEC-Bus im Selbstbau (+ Fehlerteufel 10/85)	44	07/85
Joystick	Joystick im Selbstbau	33	03/85
	Dauerfeuer-Adapter	46	08/85
RS232/V.24	Das 30-Mark-Interface (Selbstbau RS232)	29	03/85
	Genau betrachtet: Die RS232/V.24-Schnittstelle	80	05/85
Diverses	Userport-Display	36	05/85
	Reset-Taster für alle Fälle (+ Fehlert. 9/85)	130	06/85
	Aus eins mach vier (absturzfreie Betriebssystemumschaltung)	41	07/85

Hardware-Grundlagen

Stichwort	Titel	Seite	Ausgabe
Computer	Was bringt der C 128?	28	11/85
Drucker	Welcher Drucker ist der Richtige? (Grundlagen)	15	05/85
	Hammerwerke — wie funktionieren Typenraddrucker	32	06/85
	Die Alternativen: Thermo-, Tintenstrahldrucker + Plotter	24	07/85
Eingabegeräte	Versteht Sie Ihr Computer? (Wie funktionieren Eingabegeräte)	44	09/85
Floppy	Floppy oder Datasette?	129	06/85
Monitore	Wie funktionieren sie, was ist beim Kauf zu beachten?	16	12/85
	Das Kabel zum Monitor: Welche Normen gibt es?	28	12/85
Peripherie	Grafikeingabegerät: Wie funktionieren sie?	30	08/85

Hardware-Tests

Stichwort	Titel	Seite	Ausgabe
Computer	Generationswechsel: Test C 16	16	01/85
	Erster ausführlicher Test C 128 PC (Teil 1)	16	06/85
	Erster ausführlicher Test C 128, PC (Teil 2)	17	07/85
DFÜ	Marktübersicht Modems & Akustikkoppler	32	07/85
Drucker	Vergleich: Drucker unter 700 Mark (Tests und Marktübersicht)	18	05/85
	Tests und Marktübersicht Typenraddrucker	35	06/85
	Test: Brother EP 44	27	07/85
	Brother TC-600	118	08/85
	Riteman C +	133	09/85
	Panasonic KX-P1091	134	09/85
	Star SG 10C	132	09/85
	Melchers CP-80X — wie hätten Sie's denn gern?	25	10/85
	Geheimtip: Der RFI DP 165	24	10/85
	Epson GX 80 — einer für alle	26	10/85
	MPS 803 — ein Drucker für alle Gelegenheiten?	40	1/85
	Epson JX-80 das vielfarbige Druck-Genie	38	11/85
	Epson FX-85 neue Referenz	42	11/85
	SP 1000 VC — Superstar mit Haken	41	11/85
	Der NEC-P2 — das fernöstliche Wunder	159	12/85
	DMPG9 — eine solide Sache	162	12/85
	Das Doppelleben des Joystick-Ports: 10er-Tastaturen	50	09/85
	Joysticks: Test und Marktübersicht (+ Fehlerteufel 12/85)	19	11/85
	Es geht auch anders: Lightpens und Trackballs	22	11/85
EPROMer	Frisch gebrannt ist halb gespeichert (EPROM-Programmiergeräte im Test)	39	07/85
	QuickByte II — das Kraftpaket	14	10/85
Floppy/Datasette	Turbo-Floppies, zweite Generation: Speeddos plus + Prologic DOS	28	10/85
	Das große Rennen: Schnelle Bandlaufwerke	37	10/85
	Professionelle Floppylaufwerke für den C 64 (IEC-Floppies)	30	10/85
	Gut gekauft ist halb gespeichert (Marktübersicht Disketten)	38	10/85
Grafik	Die Videowerkstatt (Digitizer-Test)	32	05/85
	Digitalbilder m.d. C 64: PrintTechnik Digitizer	24	01/85
Interface	Hardware-Interface ganz weich: Test EC 64	23	01/85
	Gute Connections — Übersicht Schnittstellen	21	03/85
	Card/Print + 6 — Das Allround-Interface	20	03/85
	Das Wiesemann-Centronics-Interface	18	03/85
	Erst ein IEC-Bus öffnet Tür und Tor (+ Fehlert. 4/6-85)	24	03/85
Monitore	Marktübersicht: Monochrome Monitore	30	12/85
Musik	Trommelwirbel: Test Digital Drums	45	08/85
	Die Musikhardware zum C 64	17	09/85
Roboter	Roboter selbst gebaut (Fischertechnik)	167	10/85
Scanner	So lernt Ihr Drucker lesen	30	06/85
Speicher	Speichertuning VC 20: Test 64 KByte Karte	26	01/85
Steuern	Flottes Türmchen: MEA-Interface	116	08/85

Kurse

Stichwort	Titel	Seite	Ausgabe
Assembler	Assembler ist keine Alchimie, Teil 5	142	01/85
	Assembler ist keine Alchimie, Teil 7	124	03/85
	Assembler ist keine Alchimie, Teil 9	138	05/85
	Assembler ist keine Alchimie, Teil 10	127	07/85
	Assembler ist keine Alchimie, Teil 11	126	08/85
	Assembler ist keine Alchimie, Teil 12	109	09/85
	Assembler ist keine Alchimie, Teil 13 (Schluß)	143	10/85
C 128	Entdeckungsreise duch den C 128	42	12/85
Effektives Programieren	Müllabfuhr im Computer: Garbage Collection, Teil 1	122	01/85
	Finden mit System, eine neuartige Suchmethode, Teil 3	148	03/85
	Sortieren mit dem Computer, Teil 2	159	05/85
	Sortieren mit dem Computer, Teil 3	124	06/85
	Sortieren mit dem Computer, Teil 4	138	08/85
	Sortieren mit dem Computer, Teil 5	124	09/85
	Sortieren mit dem Computer, Teil 6 (Schluß)	150	12/85
Extern	C 64 extern — Der Weg nach draußen, Teil 1	144	08/85
	C 64 extern — Der Weg nach draußen, Teil 2	122	09/85
	C 64 extern — Der Weg nach draußen, Teil 3 (Schluß)	129	10/85
Floppy	In die Geheimnisse der Floppy eingetaucht, Teil 4	148	01/85
	In die Geheimnise der Floppy eingetaucht, Teil 5	130	03/85
	In die Geheimnise der Floppy eingetaucht, Teil 6	145	05/85
	In die Geheimnisse der Floppy eingetaucht, Teil 7 (Schluß)	116	06/85
	Directory-Manipulationen I	140	06/85
Floppy	Directory-Manipulationen II	163	10/85
Grafik	Hires 3 — 15 neue Basic-Befehle, Teil 2	136	03/85
	Hires 3 — Grafikkurs-Anwendung, Teil 3 (Schluß)	152	08/85
	Sprites ohne Geheimnisse	40	08/85
	Streifzüge durch die Grafikwelt, Teil 1	106	09/85
	Streifzüge durch die Grafikwelt, Teil 2	149	11/85
Logeleien	Logeleien, Teil 1	143	07/85
	Logeleien, Teil 2	136	08/85
	Logeleien, Teil 3 (Schluß)	115	09/85
Musik	Dem Klang auf der Spur, Teil 2	136	01/85
	Dem Klang auf der Spur, Teil 4	131	04/85
	Dem Klang auf der Spur, Teil 5	152	05/85
	Dem Klang auf der Spur, Teil 7	132	07/85
	Dem Klang auf der Spur, Teil 8	133	08/85
	Dem Klang auf der Spur, Teil 9	126	10/85
	Dem Klang auf der Spur, Teil 10 (Schluß)	157	11/85
Speicher	Memory Map mit Wandervorschlägen, Teil 3	126	01/85
	Memory Map mit Wandervorschlägen, Teil 5	144	03/85
	Memory Map mit Wandervorschlägen, Teil 7	120	06/85
	Memory Map mit Wandervorschlägen, Teil 8	140	07/85
	Memory Map mit Wandervorschlägen, Teil 9	129	08/85
	Memory Map mit Wandervorschlägen, Teil 10	112	09/85
	Memory Map mit Wandervorschlägen, Teil 11	133	10/85
	Memory Map mit Wandervorschlägen, Teil 12	145	11/85
	Memory Map mit Wandervorschlägen, Teil 13	146	12/85
Sprachen	Basic ist out — es lebe Forth	43	01/85
VC 20	Der gläserne VC 20, Teil 4	130	01/85
	Der gläserne VC 20, Teil 6 (Schluß)	155	03/85

Software-Tips

Stichwort	Titel	Seite	Ausgabe
C 128	Erste Fragen und Antworten zum C 128	14	09/85
	Fragen und Antworten zum 128er	20	10/85
	Fragen und Antworten zum 128er	40	12/85
Drucker	Der MPS 802 lernt Deutsch	30	05/85
	Centronics-Interface für jeden Bedarf	78	07/85
Textverarbeitung	Software Corner — professionelle Programme richtig eingesetzt (Vizawrite-Tips)	174	12/85
Tips & Tricks	Autoboot beim C 64	86	03/85
	Verbindungsfreundlich (Parallelschnittstelle des VC 20)	91	03/85
	Undefinierte Opcodes des 6502	84	03/85
	Durch POKEs zum Erfolg (Spiele-POKEs)	83	03/85
	Tips und Erweiterungen zu Hi-Eddi und Simons Basic	88	03/85
	Basic-Befehle im Griff	79	05/85
	Durch POKEs zum Erfolg: Spiele-POKEs	78	06/85
	Formatierte Eingabe	148	06/85
	Hi-Text (Text in Hires)	70	08/85
	Verbotene Variablen	66	09/85
	Verschiedene Routinen für Anfänger und Profis (+ Fehlerteufel 12/85)	88	11/85
	Der Trick mit dem Joystick (Joystickabfrage)	24	11/85
	Verschiedene Tips für Anfänger und Fortgeschrittene	106	12/85

Software-Grundlagen

Stichwort	Titel	Seite	Ausgabe
Assembler	Assembler? Assembler! (Einführung)	32	01/85
	Assembler-Bedienung leicht gemacht, Teil 1	169	12/85
DFÜ	Der erste Kontakt mit DFÜ	40	06/85
	Die Netze der Post: Btx, Datex-P, Telebox	46	06/85
	DFÜ — Was ist das?	44	06/85
	Mailbox für Anfänger	30	07/85

Listings zum Abtippen

Software-Tests

So machen's andere

```
9498 : 8e 95 a5 06 20 8e 95 a9  3b
94a0 : 85 2c a9 c6 20 8e 95 a9  9f
94a8 : 01 20 8e 95 28 b0 05 a9  7f
94b0 : 58 20 8e 95 a9 4c 20 8e  09
94b8 : 95 a5 0a 20 8e 95 a5 0b  e9
94c0 : 20 8e 95 d0 0f a2 ef bd  e8
94c8 : e4 96 20 8e 95 e8 d0 f7  a5
94d0 : a9 47 85 0e 38 20 a1 95  76
94d8 : 84 e4 a0 16 18 a6 07 bd  08
94e0 : 0e 98 65 0e 91 e0 a0 46  85
94e8 : 18 69 01 91 e0 a9 c4 fd  92
94f0 : 0e 98 38 e5 0e a0 11 91  62
94f8 : e0 a0 34 18 69 47 91 e0  11
9500 : a5 0e 49 ff a8 18 a5 0e  fd
9508 : ca 10 1b 69 64 99 6e 03  a1
9510 : 69 07 99 67 03 a6 03 ca  57
9518 : 86 68 8a 99 76 03 98 69  5d
9520 : bc 8d a2 03 10 58 ca 10  bb
9528 : 17 69 68 99 5b 03 69 07  c3
9530 : 99 54 03 a5 03 8d b9 03  f2
9538 : 98 69 aa 8d a1 03 10 3e  d0
9540 : ca 10 17 69 85 99 23 03  bd
9548 : a5 03 99 11 03 98 69 91  b5
9550 : 8d 92 03 e9 21 8d 94 03  fb
9558 : 10 24 69 80 99 f0 00 a5  51
9560 : a7 8d 6a 01 a5 a9 99 ff  97
9568 : 00 a5 a8 99 03 01 98 69  06
9570 : 60 8d 6c 01 a5 b3 8d 5d  bb
9578 : 01 a5 b4 8d 61 01 a2 ee  b1
9580 : a4 e4 bd f6 96 20 8e 95  b4
9588 : e8 d0 f7 84 e4 60 91 e0  c0
9590 : c8 d0 04 e6 e1 e6 e5 c4  15
9598 : e5 d0 03 ee 78 03 4c 76  42
95a0 : 93 a5 07 2a aa 84 0c 38  7c
95a8 : bd f6 97 e5 0c 85 e2 bd  77
95b0 : fe 97 e9 00 85 e3 bd 06  6f
95b8 : 98 aa b1 e2 20 8e 95 ca  d0
95c0 : d0 f8 60 20 b2 01 29 3f  7f
95c8 : 85 ff 8a 0a b0 38 30 17  ec
95d0 : e0 3f d0 05 20 b2 01 85  cb
95d8 : ff 20 b2 01 20 bb 01 c6  26
95e0 : ff d0 f6 a5 ad d0 dc a2  d4
95e8 : 02 38 a5 ae e5 ff 85 fc  b4
95f0 : a5 af e9 00 85 fd b1 fc  f0
95f8 : e6 fc d0 02 e6 fd 20 bb  27
9600 : 01 ca d0 f2 f0 dd 10 08  47
9608 : 20 b2 01 a6 ff 85 ff 2c  1b
9610 : a2 03 b0 d6 86 f7 e8 86  14
9618 : f8 a2 ff d0 53 20 b2 01  7e
9620 : 29 3f 85 5c 85 5d 84 5f  ea
9628 : 8a f0 45 0a 90 51 30 08  21
9630 : a2 04 86 5c c9 7e d0 05  df
9638 : 20 b2 01 85 5d 20 b2 01  46
9640 : c5 f8 b0 09 c9 ff b0 05  38
9648 : 85 5f 20 b2 01 a6 5c 85  9d
9650 : 5e 38 a5 ae e5 5e 85 5e  2e
9658 : a5 af e5 5f 85 5f b1 5e  11
9660 : 20 aa 01 e6 5e d0 02 e6  35
9668 : 5f ca d0 f2 a6 5d f0 0b  ee
9670 : b1 ac 20 b4 01 20 aa 01  d4
9678 : ca d0 f5 a5 ad d0 9e 30  19
9680 : 07 4a 84 5d a2 02 d0 c7  86
9688 : 20 b2 01 a2 03 d0 c0 e6  1d
9690 : f7 d0 0d e6 f8 d0 09 86  57
9698 : f7 86 f8 a2 ff d0 67 20  c9
96a0 : b2 01 29 3f 85 5c 85 5d  11
96a8 : 84 5f 84 60 8a f0 57 0a  ab
96b0 : 90 63 30 0a a2 04 86 5c  5c
96b8 : c9 7e d0 14 f0 0d a5 5c  3e
96c0 : c9 05 b0 07 85 60 20 b2  5a
96c8 : 01 85 5c 20 b2 01 85 5d  ab
96d0 : 20 b2 01 38 e5 f8 90 02  fd
96d8 : d0 05 86 5f 20 b2 01 a6  a1
96e0 : 5c 85 5e 38 a5 ae e5 5e  c2
96e8 : 85 5e a5 af e5 5f 85 5f  2a
96f0 : b1 5e 20 a2 01 e6 5e d0  8f
96f8 : 02 e6 5f ca d0 f2 c6 60  1f
9700 : 10 ee a6 5d f0 0b b1 ac  64
9708 : 20 b4 01 20 a2 01 ca d0  c6
9710 : f5 a5 ad d0 8a 30 07 4a  38
9718 : 84 5d a2 02 d0 c3 20 b2  45
9720 : 01 a2 03 d0 bc e6 f7 d0  d2
9728 : 15 a4 f8 c0 ff f0 02 e6  43
9730 : f8 a0 00 f0 09 b1 ac c9  fb
9738 : ff f0 07 20 97 01 a5 ad  e9
9740 : d0 f3 20 b4 01 20 b2 01  86
9748 : 85 a0 20 b2 01 65 ac 85  75
9750 : a1 a5 ac 85 a2 a5 ad 85  b9
9758 : a3 20 97 01 a5 ac c5 a1  2b
9760 : d0 f7 c6 a0 f0 d8 a5 a2  a3
9768 : 85 ac a5 a3 85 ad b0 e9  7d
9770 : 20 b2 01 c9 ff f0 0b c9  aa
9778 : ff d0 19 20 b2 01 b1 ac  7d
9780 : d0 03 b3 ac 98 20 bb 01  d0
```

```
9788 : ca d0 fa 01 08 0c 08 c3  22
9790 : 07 9e 32 30 36 32 ff 00  6e
9798 : 00 00 78 a0 79 b9 46 08  59
97a0 : 99 46 01 88 d0 f7 84 01  8f
97a8 : 84 ac 84 ad a2 04 b5 aa  d0
97b0 : d0 02 d6 ab d6 aa ca ca  30
97b8 : d0 f4 b1 ae 91 ac a9 c0  eb
97c0 : c5 ae a9 08 e5 af 90 e4  30
97c8 : a9 01 85 ae a9 08 85 af  79
97d0 : 4c 47 01 a5 ae 85 2d a5  cc
97d8 : af 85 2e c6 01 20 59 a6  72
97e0 : 58 4c ae a7 b3 ac e6 ac  94
97e8 : d0 02 e6 ad 60 91 ae e6  44
97f0 : ae d0 02 e6 af 60 97 15  ea
97f8 : 14 7f c3 e7 35 42 96 97  a9
9800 : 96 96 95 95 97 97 7e 20  6a
9808 : 6b 18 24 2d 0d 49 bb d6  e6
9810 : 08 03 ff ff ff ff ff ff  9a
9818 : a0 50 20 12 84 a6 09 8e  e9
9820 : 11 d0 20 48 94 a2 59 bd  ea
9828 : 9b 98 9d 50 05 ca 10 f7  58
9830 : a6 ac a5 ad a4 05 10 2c  56
9838 : a9 58 8d 12 08 a4 b0 b9  8f
9840 : 0e 08 99 14 08 88 d0 f7  33
9848 : ad 16 08 38 e9 12 8d 16  9b
9850 : 08 ad 33 08 e5 c1 a8 b9  7f
9858 : 14 08 99 26 08 88 d0 f7  93
9860 : a2 27 a9 08 86 fa 8e 7e  79
9868 : 03 85 fb 8d 82 03 a4 07  bf
9870 : a9 c0 85 fc a2 04 b9 95  d7
9878 : 98 85 fe b9 98 98 85 ff  2e
9880 : a0 ea 86 fd b1 fe 91 fc  4a
9888 : c8 d0 f9 e6 ff e8 e0 07  ec
9890 : d0 f0 4c 6f 05 16 56 36  a0
9898 : 9c 8c 8b 84 fc a9 08 85  36
98a0 : fd 88 84 fe 84 ff 20 50  4c
98a8 : 04 a9 01 ed 74 04 85 fc  f6
98b0 : a9 08 ed 75 04 85 fd 4c  84
98b8 : e2 04 e6 01 a0 fe 84 fe  88
98c0 : c8 84 ff a5 fb 85 fd a5  ae
98c8 : fa 85 fc 20 50 04 a5 fa  7a
98d0 : e5 ae 85 fe a5 fb e5 af  7f
98d8 : 85 ff 88 c6 fb ad ff ff  85
98e0 : 91 fb c8 84 5d 84 58 b1  f0
98e8 : fe 8d ff ff a5 fe 85 fc  0f
98f0 : 85 5e 18 a5 ff ff ff ff  5f
98f8 : ff ff ff ff ff ff ff ff  f7
9900 : ff ff ff ff ff ff ff ff  ff
9908 : ff ff ff ff ff ff ff ff  07
9910 : ff ff ff ff ff ff ff ff  0f
9918 : 3d 27 29 be 3b 0a 55 53  0b
9920 : 41 4b d6 2b 25 27 d1 62  b9
9928 : 1f 16 50 0a 1c 4e 18 48  cd
9930 : 04 0d 42 52 a4 03 8c 58  db
9938 : f0 26 3c 79 26 88 7a 32  6f
9940 : 66 2c 3a 95 6e 22 ed 2a  02
9948 : 7c 77 95 12 5c 10 1a 0e  f2
9950 : ec 4b c4 14 c8 1a 00 43  79
9958 : de e2 f0 fc be f0 bd f4  d7
9960 : cc a3 e2 ee cf e6 8a e2  b8
9968 : 1c 8d 78 b1 e6 e0 81 18  4b
9970 : cc fc 02 9b 4e c4 e0 91  60
9978 : 6c 8f 35 be 94 66 64 12  03
9980 : 6e a4 a8 18 90 f7 ea a2  27
9988 : 96 9c 2a 52 c5 aa b0 c3  3d
9990 : 4c d1 28 c7 51 0b 80 52  dc
9998 : 9d d3 08 2f fe 77 65 49  db
99a0 : b1 3e 69 2a 07 7f 73 76  37
99a8 : 97 47 4b e1 df 4d 31 17  4d
99b0 : 49 61 6b 53 3f 1f 00 42  61
99b8 : b6 85 a9 db 9f 25 36 79  06
99c0 : 31 3d 78 61 3b fe e6 60  e2
99c8 : 5c 5e 58 5a 50 58 78 3e  db
99d0 : fa 47 48 4a 44 46 40 42  c5
99d8 : bc be b8 bb b4 b6 b0 b2  c2
99e0 : ec 3e 50 2b 6f 55 a2 5f  50
99e8 : 98 98 93 c9 35 96 80 d2  9a
99f0 : 1e 9f 93 cb c1 15 39 39  58
99f8 : 31 1f f0 f0 f8 ee d0 c2  e3
9a00 : b8 7e 7c 72 24 2c 30 30  2a
9a08 : 26 22 d1 c3 f5 9d 89 8f  be
9a10 : 93 af ab 4f 4f 4b 4f 53  83
9a18 : ab e1 cf c7 38 26 10 7a  8b
9a20 : 62 7c 40 80 9c 98 fa c6  e8
9a28 : 19 11 0f 05 35 31 23 23  de
9a30 : 49 5d 5f 53 61 71 79 81  f5
9a38 : fd e7 dd bd a7 9d 9d 87  45
9a40 : 95 76 46 5a 58 26 2a 3a  c1
9a48 : 02 32 28 fa f0 e6 e6 98  e0
9a50 : 9a ac ba ba b0 ba 6d 75  c8
9a58 : 85 ff d7 c9 c5 33 1d 0d  90
9a60 : 67 67 79 7f 45 43 49 4d  f7
9a68 : 6f 6f 8e 8e 8a be a6 a6  8b
9a70 : b2 ca e8 e0 ea 00 0e 0e  e0
```

```
9a78 : 64 60 5e 52 5a 36 10 f1  6a
9a80 : ed c7 d5 83 57 40 b8 99  c4
9a88 : 4d ee 2e 4c e4 54 55 78  99
9a90 : e8 95 6e 97 bd 37 66 5a  b5
9a98 : 5e a9 a1 99 ad b4 1e 85  6a
9aa0 : 29 d9 9e 23 63 9e 0c 45  a8
9aa8 : 9e 97 27 b2 1f 7c 69 7c  a6
9ab0 : 71 e7 af 81 72 d3 cb 6f  05
9ab8 : f5 34 3c 3d 15 8e e4 b6  45
9ac0 : 11 f7 65 ea e3 a0 01 32  2f
9ac8 : 21 d7 5d 6c c1 78 66 6a  08
9ad0 : fb 02 ea ec 20 ce df de  da
9ad8 : f6 80 af 42 cb 90 80 8c  9f
9ae0 : 75 2b d1 fd b9 f4 1f c8  70
9ae8 : de b3 d2 84 91 40 dd 4e  14
9af0 : 90 64 8b 7c 15 05 6d c4  dd
9af8 : f5 45 f5 3d 95 af 0d f3  a8
9b00 : 46 7a 8b db c6 7d 6a 6f  c2
9b08 : d9 f8 49 f5 ac b9 7a f8  e3
9b10 : 34 ed e0 0d 4c 83 20 99  a9
9b18 : f2 ff 7c 0a 14 b9 60 aa  50
9b20 : c5 c7 d1 a8 45 86 df b1  be
9b28 : 6f 5d e4 a6 34 11 0a 59  fa
9b30 : 8f fa 55 74 28 78 2f 18  53
9b38 : 7a 1f b1 06 77 a0 0e 71  07
9b40 : b0 24 7c ba 9e 60 c7 23  cb
9b48 : f3 13 6b 7e 5b bc d8 58  1f
9b50 : 4e 05 fd 10 50 ba d5 ae  32
9b58 : 40 c7 ba 2b 23 d0 92 17  c1
9b60 : 33 3b 1f bf 55 8e 08 12  ff
9b68 : 98 8a f8 0c db d9 d4 f7  d5
9b70 : 42 12 5a fb a1 bf 02 c9  85
9b78 : b5 fa 20 4a 2c 07 9a 51  04
9b80 : 9c 01 24 42 85 ae 10 74  e5
9b88 : 05 10 36 e8 39 1f 41 09  e4
9b90 : be ea 81 be 24 37 24 09  9a
9b98 : 95 94 6c e1 fb 60 5a 4e  97
9ba0 : 2f be be e3 fc 71 aa e6  2e
9ba8 : 35 41 47 ae 11 6a c5 ae  fe
9bb0 : 8b 1d ff 44 78 a1 e9 bb  06
9bb8 : 23 22 94 f0 f2 d2 c3 92  29
9bc0 : a1 ad 8a 2a ca 6a db 55  3a
9bc8 : 77 db 8f 12 b6 f6 27 a0  54
9bd0 : 5d f7 e2 53 ec b9 9a 8f  72
9bd8 : d4 4e e0 94 30 9b 74 39  c2
9be0 : 30 7d 18 75 06 68 35 d4  a6
9be8 : b7 2f 1e 79 c1 df 7c 6c  d3
9bf0 : 62 f2 ee f2 74 5b 05 63  e2
9bf8 : 19 03 07 17 21 13 cb 32  76
9c00 : f1 36 76 0d a5 01 07 7a  bf
9c08 : ec 5f 77 db f8 59 2f b8  86
9c10 : 6c ab c3 75 0c b7 cb e2  65
9c18 : 74 22 8c 72 f9 1e d1 fa  dc
9c20 : 55 60 4d 9a 97 d8 7e 43  0d
9c28 : e7 83 df 85 ce 10 83 5e  b2
9c30 : f1 b9 b7 33 bf 98 71 d3  80
9c38 : bc b0 c9 11 c2 f6 aa 1c  a8
9c40 : 95 5f 31 e5 64 53 ac 80  22
9c48 : d4 77 65 9f c1 e5 7e f9  5e
9c50 : fc 1a 35 39 f6 34 8e 3a  8e
9c58 : 62 6b 08 76 64 7f 55 0b  ee
9c60 : 10 2b c8 bc ff 9b 1a 3d  8f
9c68 : eb cd 0f 7c af 91 2d f0  ab
9c70 : e8 a4 0d c6 0b b7 d3 2e  e1
9c78 : cc e1 5d a2 81 12 c3 79  8b
9c80 : e7 53 a7 35 ba c7 5b 8c  12
9c88 : 65 d6 84 f8 3f 7d fa b7  d4
9c90 : 13 6c 88 5b 7e d2 59 78  3c
9c98 : 6c bd 64 a3 5c 09 4c d0  51
9ca0 : c9 78 77 23 41 b1 27 79  19
9ca8 : ad 84 c8 aa f6 70 39 ed  d2
9cb0 : 7c 27 ac 34 d1 88 17 fd  2b
9cb8 : bf 6a 61 01 49 60 1e ab  8c
9cc0 : 0a 19 6f f7 c5 07 37 66  70
9cc8 : cb 4e b8 bd 95 f6 6e b7  da
9cd0 : f0 75 d8 c5 cc 2a 4f 2b  1b
9cd8 : e5 32 68 ef 57 25 ff ff  8d
9ce0 : ff ff ff ff ff ff ff ff  df
9ce8 : ff ff ff ff ff ff ff ff  e7
9cf0 : ff ff ff ff ff ff ff ff  ef
9cf8 : ff ff ff ff ff ff ff ff  f7
9d00 : 85 fd 65 03 90 05 a9 63  dc
9d08 : 85 5e 98 85 5f b1 5e 85  9b
9d10 : 60 84 61 84 67 a0 00 ad  72
9d18 : ff ff 91 5e c8 d1 fc f0  38
9d20 : 07 c8 d0 f9 e6 fd d0 f5  8c
9d28 : 98 18 65 fc 85 fc 90 02  4c
9d30 : e6 fd 38 e5 fe 85 65 a7  e1
9d38 : fd e5 ff c5 03 b0 25 85  36
9d40 : 66 8a 25 fc c9 ff f0 1c  6d
9d48 : a2 00 a5 60 81 5e a0 01  ef
9d50 : b1 fe d1 fc d0 11 c8 d0  ef
9d58 : f7 a5 fd 85 62 a5 ff 85  b0
9d60 : 64 4c b5 06 4c ff 06 a6  43
```

```
9d68 : 61 d0 aa a6 fd e8 d0 0b   31
9d70 : a5 fc 49 ff 85 57 c4 57   ba
9d78 : 90 01 a8 a2 01 c0 02 90   46
9d80 : 94 d0 0e a5 65 29 c0 d0   f9
9d88 : 8c a5 66 05 5d d0 86 f0   79
9d90 : 58 c0 04 b0 0c a5 66 d0   89
9d98 : 66 a5 5d 29 c0 d0 60 f0   43
9da0 : 48 d0 23 a5 66 d0 15 18   3f
9da8 : 98 65 fe a5 ff 69 00 49   45
9db0 : ff c5 68 90 03 18 a5 68   16
9db8 : 65 65 90 01 ca a5 5d c9   f7
9dc0 : 3f b0 26 e8 d0 23 98 e9   5a
9dc8 : 03 aa a5 66 d0 15 18 98   9e
9dd0 : 65 fe a5 ff 69 00 49 ff   d9
9dd8 : c5 68 90 03 18 a5 68 65   71
9de0 : 65 90 01 ca c0 40 90 01   79
9de8 : ca e4 67 90 12 d0 04 c4   51
9df0 : 58 b0 0c 86 67 84 58 a5   bb
9df8 : 65 85 59 a5 66 85 5a 4c   bf
9e00 : bf 05 b1 fe d1 fc d0 0c   ee
9e08 : c8 d0 f7 e6 ff e6 fd e8   14
9e10 : e0 05 90 ee a5 62 85 fd   f4
9e18 : a5 64 85 ff 38 a9 ff e5   ed
9e20 : fc 85 62 a9 ff e5 fd 85   df
9e28 : 64 c4 62 8a e5 64 90 04   a4
9e30 : a6 64 a4 62 e0 05 f0 11   9a
9e38 : c4 58 8a e5 61 90 c0 d0   c7
9e40 : 04 c4 58 f0 ba 86 61 d0   e1
9e48 : ac 86 61 84 58 a5 66 85   78
9e50 : 5a a5 65 85 59 a0 ff a6   6f
9e58 : 61 d0 23 a4 58 f0 78 88   7f
9e60 : 88 d0 0b a5 59 a2 02 84   84
9e68 : 5d 20 73 04 d0 79 88 d0   cf
9e70 : 0d a5 59 20 73 04 a5 5d   53
9e78 : 09 40 a2 03 d0 e9 a6 5a   56
9e80 : d0 16 18 a5 fe 65 58 a5   de
9e88 : ff 65 61 49 ff c5 68 90   ac
9e90 : 03 18 a5 68 65 59 90 06   85
9e98 : a5 59 20 73 04 8a 20 73   5c
9ea0 : 04 88 d0 11 a5 5d c9 3f   29
9ea8 : 90 05 20 73 04 a9 3f 09   ce
9eb0 : 80 a2 04 d0 b2 a5 5d 20   aa
9eb8 : 73 04 a0 00 a5 58 a6 61   d0
9ec0 : d0 04 c9 40 90 0c 48 20   d8
9ec8 : 73 04 68 d0 01 ca 8a a6   4f
9ed0 : 58 24 aa 09 c0 d0 90 ad   36
9ed8 : ff ff 20 73 04 a2 01 e6   75
9ee0 : 5d d0 04 98 20 73 04 8a   7c
9ee8 : 18 65 fe 85 fe a5 ff 65   0b
9ef0 : 61 85 ff 25 fe c9 ff b0   58
9ef8 : 07 a5 60 91 5e 4c 9a 05   d9
9f00 : a5 5d 8d 8d 03 4c 50 05   47
9f08 : ff ff ff ff ff ff ff ff   07
9f10 : ff ff ff ff ff ff ff ff   0f
9f18 : ff ff ff ff ff ff ff ff   17
9f20 : ff ff ff ff ff ff ff ff   1f
9f28 : ff ff ff ff ff ff ff ff   27
9f30 : ff ff ff ff ff ff ff ff   2f
9f38 : ff ff ff ff ff ff ff ff   37
9f40 : ff ff ff ff ff ff ff ff   3f
9f48 : ff ff ff ff ff ff ff ff   47
9f50 : ff ff ff ff ff ff ff ff   4f
9f58 : ff ff ff ff ff ff ff ff   57
9f60 : ff ff ff ff ff ff ff ff   5f
9f68 : ff ff ff ff ff ff ff ff   67
9f70 : ff ff ff ff ff ff ff ff   6f
9f78 : ff ff ff ff ff ff ff ff   77
9f80 : ff ff ff ff ff ff ff ff   7f
9f88 : ff ff ff ff ff ff ff ff   87
9f90 : ff ff ff ff ff ff ff ff   8f
9f98 : ff ff ff ff ff ff ff ff   97
9fa0 : ff ff ff ff ff ff ff ff   9f
9fa8 : ff ff ff ff ff ff ff ff   a7
9fb0 : a0 00 98 18 79 92 83 c8   45
9fb8 : c0 24 d0 f8 59 91 83 88   1f
9fc0 : d0 fa 49 9a d0 ef 85 aa   ab
9fc8 : a9 0b 85 ab 85 af a0 c6   b3
9fd0 : 84 ae a9 18 85 ac a9 9c   b6
9fd8 : 85 ad a2 04 88 b1 ac 49   b8
9fe0 : a5 45 ad 85 03 98 47 03   5c
9fe8 : b0 02 49 80 91 aa 98 d0   6e
9ff0 : eb c6 ab c6 ad ca d0 e4   40
9ff8 : a2 fe 9a 86 03 4c 0e 08   6c
```

Listing. (Schluß)

Von Packern und Kompressoren (1)

Grafiken oder lange Programme belegen auf der Diskette oder im Computer wertvollen Speicherplatz. Das Komprimieren eines Programms schafft hier Abhilfe. Wir zeigen Ihnen die Grundlagen solcher Algorithmen in der Praxis.

Dieser Artikel soll über verschiedene Möglichkeiten informieren, wie man Speicherbereiche, egal ob sie mit Basic-, Maschinensprache-Programmen oder Bildern belegt sind, komprimieren kann. Am Ende soll ein Programm mit zwei Packern stehen, das in der Lage ist, den Speicherplatz auf verschiedene Arten zusammenzufassen und damit auch die Stärken und Schwächen der beiden Algorithmen aufzeigen kann. Dieses Programm soll aber nur der »Rohbau« sein, den Sie selbst ausgestalten und erweitern können. Obwohl es möglich ist, Packer in Basic zu programmieren, ist dieses aus technischen Gründen nicht empfehlenswert. Deshalb werde ich versuchen, die Arbeitsweise der Kompressoren mit Hilfe der Maschinensprache zu erklären. Sollten Sie Assembler noch nicht beherrschen, kann Sie dieser Artikel vielleicht ermutigen, sich mit dieser Sprache zu beschäftigen. Sie sieht wesentlich schwerer aus, als sie ist. Doch nun endlich zu den Packern.

Die erste Idee wie man den Speicherplatz zusammenfassen kann erhalten Sie, wenn Sie sich ein Grafikbild mit Hilfe eines Maschinensprache-Monitors im Speicher anschauen. Dort finden Sie Bytefolgen wie in Bild 1.

Sie erkennen eine große Anzahl gleicher Bytes hintereinander. Es fällt sofort auf, daß zum Beispiel nicht 32mal eine $FF (255 dezimal) im Speicher stehen muß, sondern nur die Anzahl der Bytes und der Code, der den Speicher gefüllt hat. In diesem Fall würde es ausreichen, eine 32 ($20) und dann eine 255 ($FF) im Speicher stehen zu haben. Aber so einfach ist das auch wieder nicht. Wie sollte der Computer unterscheiden, ob nun 32mal der Wert 255 gemeint ist, oder ob man einfach zwei Daten, ein Datum mit dem Wert 32 und ein Datum mit 255, gespeichert hat.

Deshalb muß man vor die beiden Zahlen einen »Erkennungs-Code« setzen. Hierbei sollte man eine nicht sehr häufig auftretende Zahl verwenden. Sowohl in den folgenden Beispielen als auch in dem Programm benutze ich die Zahl $DA. Aus unseren 32 Byte wäre somit ein 3-Byte-Ausdruck geworden »DA 20 FF«.

```
2000  00 00 00 00 00 00 00 00
2008  00 00 00 00 00 00 00 00
2010  00 00 00 00 00 00 00 00
2018  00 00 00 00 00 00 00 00
2020  60 60 60 60 60 60 60 00
2028  00 00 00 00 00 00 00 00
2030  FF FF FF FF FF FF FF FF
2038  FF FF FF FF FF FF FF FF
2040  FF FF FF FF FF FF FF FF
2048  FF FF FF FF FF FF FF FF
2050  00 00 00 00 00 00 00 00
2058  00 00 00 00 00 00 00 00
2060  00 00 00 00 00 00 00 00
2068  00 00 00 00 00 32 32 32
2070  32 32 32 32 32 32 32 32
2078  32 32 32 32 32 32 32 32
```

Bild 1. So sehen Grafiken im Speicher aus: Viele Folgen gleicher Bytes

Hier erkennen Sie auch die Grenze dieses Algorithmus.

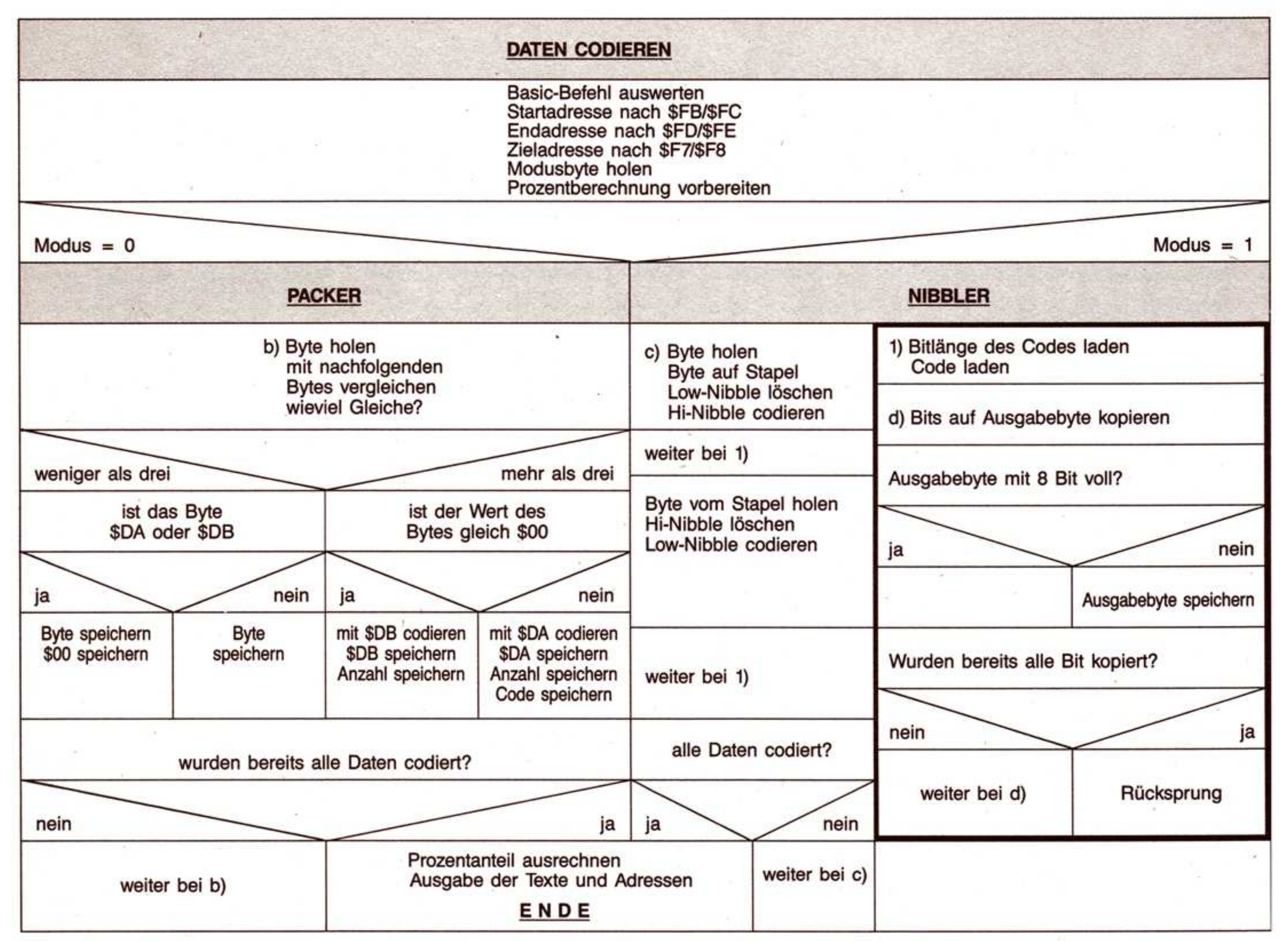


Bild 2. Das Flußdiagramm veranschaulicht den Pack-Algorithmus des dokumentierten Maschinenspracheprogramms

Wenn also weniger als drei gleiche Bytes hintereinander stehen, kann man die Bytes nicht komprimieren.

Was passiert nun aber, wenn man folgende Codefolge im Speicher stehen hat?

```
C500  80 80 80 80 80 80 DA FF
C508  FF FF FF FF FF FF FF FF
```

Von C500 bis C505 steht der Wert $80. Dieses könnte der Computer zusammenfassen zu »DA 06 80«. Nun folgt das Datum $DA. Dieses würde der Computer nicht zusammenfassen, sondern einfach übernehmen. Daraus ergibt sich $DA. Bei den nächsten 9 Byte kann man hingegen wieder komprimieren und erhält »DA 09 FF«. Das ergibt zusammen:

```
C600  DA 06 80 DA DA 09 FF
```

Nun müßte bei dem Entpacken aber wieder das Original herauskommen. So beginnt der Computer und hat als ersten Code ein $DA. Also wurde komprimiert. Da die folgenden Codes $06 und $80 sind, ist die Zuordnung eindeutig. Dann folgt wieder ein $DA. Also wurde folglich wieder komprimiert (oder?).

Die folgenden Bytes enthalten die Anzahl ($DA) und den Code ($09). Also müßte vorher 218mal ($DA entspricht 218 dezimal) eine $09 im Speicher gestanden haben. Da diese Aussage nicht richtig ist, müssen wir unser Gedankengebäude noch einmal überdenken. Des Rätsels Lösung ist eigentlich einfach. Man komprimiert einfach auch den Code $DA. Auf den ersten Blick erscheint dieses unsinnig (warum und wie kann man ein einzelnes Byte zusammenfassen), aber nur so kann der Computer erkennen, daß dieses nicht das Erkennungszeichen für den Compactor ist, sondern ein einzelner Code. Man speichert also das Codezeichen $DA, dann die Anzahl 01 und schließlich noch den »zusammengefaßten« Wert $DA. Ein weiteres Byte kann man nun noch sparen, wenn man das letzte Datum dieser Zusammenfassung wegläßt. Da nie weniger als drei gleiche Bytes komprimiert werden, muß das codierte Byte ein $DA gewesen sein, wenn die Anzahl der gleichen Codes kleiner als 3 ist.

Aus unserem Beispiel

```
C500  80 80 80 80 80 80 DA FF
C508  FF FF FF FF FF FF FF FF
```

hätte man also folgendes erhalten:

```
C600  DA 06 80 DA 01 DA 09 FF
```

Diese Bytekombination kann man eindeutig auf das Original zurückführen.

Viele Programme – besonders aber Grafiken – bestehen aus einer großen Anzahl von Nullen. So werden bei hochauflösenden Bildern freie Flächen oft als Folge von $00 im Speicher abgelegt.

Deshalb verwendet man bei den Packern noch einen weiteren Erkennungscode (im folgenden ist es der Wert $DB), der allein bei auftretenden Nullen eingesetzt wird. Da dieses ausschließlich bei $00 geschieht, braucht man das letzte der 3 Byte, das den zusammengefaßten Code enthält, nicht mehr anzugeben. Dieses ergibt eine weitere Einsparung. Allerdings muß man hier aufpassen, denn man muß auch das zweite Erkennungsbyte, wenn es im Original auftauchen sollte, »zusammenfassen«. An einem weiteren Beispiel soll das gezeigt werden:

```
C500  E0 E0 E0 E0 E0 DA 00 00
C508  00 00 00 00 1F 1F 1F 1F
C510  1F DB 00 00 00 00 00 00
```

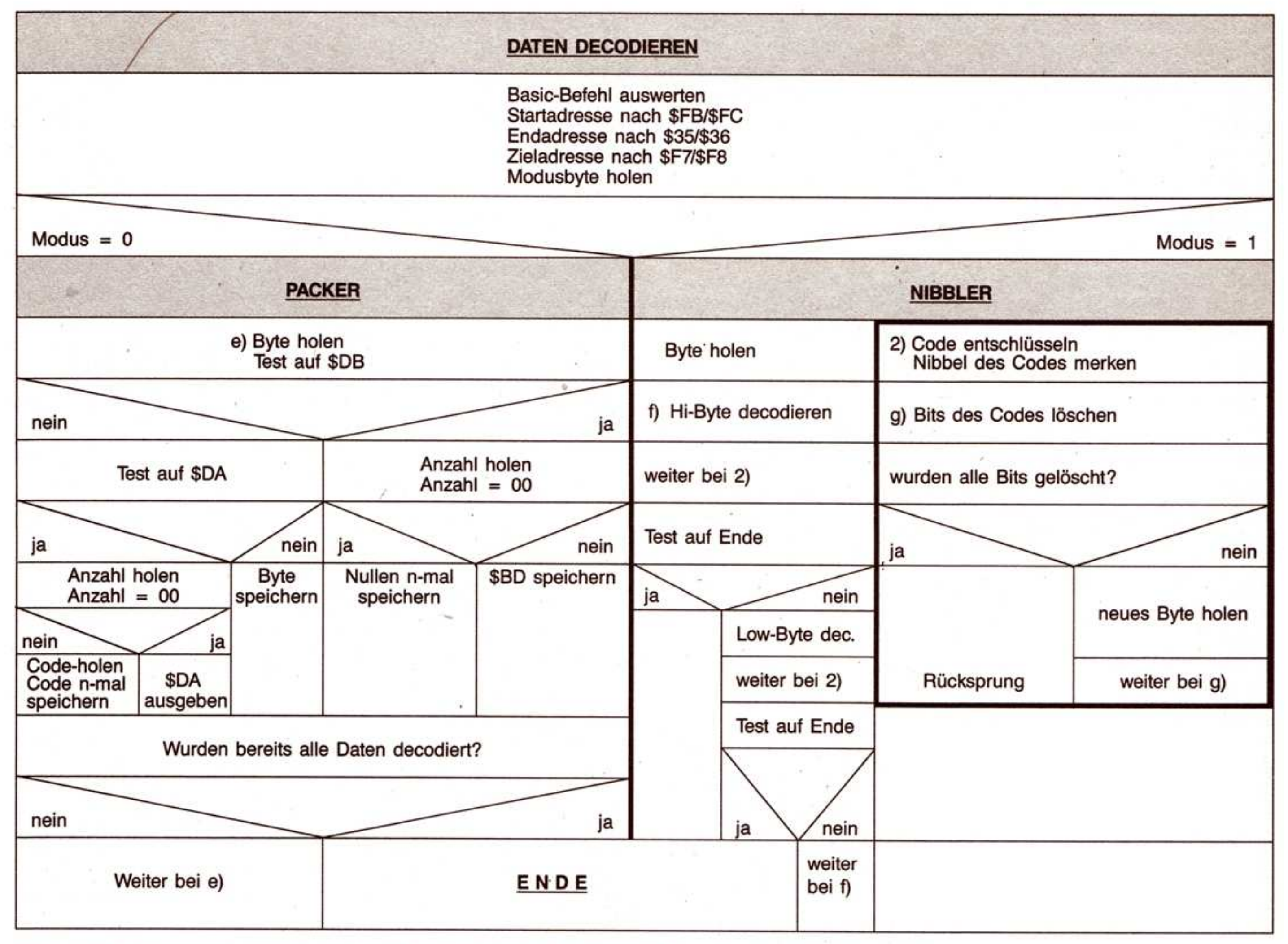


Bild 3. Flußdiagramm für den individuellen Einsatz: So werden die Daten wieder entpackt

Obwohl hier die Codes nicht danach aussehen, daß man sie besondes gut zusammenfassen kann, ist hier eine Verkürzung um mehr als ⅓ der Originallänge mit einem so einfachen Packer möglich.

```
C600  DA 05 E0 DA 01 DB 06 DA
C608  05 1F DB 01 DB 06
```

Man kann deutlich erkennen, daß Codefolgen, die ungleich Null sind, 3 Byte verbrauchen. Die Erkennungs-Codes selbst, wenn sie im Original vorkommen, benötigen 2 Byte und Nullen ebenfalls 2 Byte. Natürlich kann man die Methode, die man für die Nullen verwendet, auch auf andere Zahlen ausdehnen, aber dies wird, wenn man es übertreibt, schnell ineffektiv. In diesen Fällen müssen dann ja auch weitere Erkennungscodes verwendet werden, die man dann ebenfalls wieder aussortieren und gesondert zusammenfassen muß. Das kann den Spareffekt stark mindern. Man muß also einen Kompromiß zwischen der Anzahl der Erkennungs-Codes und der Speicherplatzersparnis finden.

Sicherlich hängt die Wahl des Verfahrens auch von dem zu komprimierenden Speicherinhalt ab. So ist es sinnvoll, neben der Null auch noch $FF besonders zu packen, wenn die zu komprimierenden Daten (wie Grafiken) hauptsächlich aus $00 und »$FF« bestehen. Man kann deshalb sagen, daß man sich für jede Situation seinen eigenen Packer zusammenstellen sollte. Aus diesem Grund finden Sie neben dem ausführlich dokumentierten Maschinensprachprogramm noch ein Flußdiagramm für beide Packer- und Entpackertypen (Bild 2 und 3).

Bevor Sie sich aber Ihren eigenen Compactor zusammenstellen, sollten Sie sich mit dem Quellcode des Programms (Listing 1) vertraut machen. Den bereits assemblierten und somit lauffähigen Code des Packers finden Sie im Programm »PACKER.OBJ« (Listing 2.) Geladen wird das Programm mit

```
LOAD "PACKER.OBJ",8,1
```

Danach muß NEW eingegeben werden. Anschließend lädt man das zu packende Programm

Die Syntax der Befehle lautet:

```
SYS 49152, Adr1, Adr2, Adr3, Mode
```

Zum Packen lauten die Adressen:

Adr 1 = Startadresse der zu komprimierenden Daten
Adr 2 = Endadresse der Daten
Adr 3 = Zieladresse der gepackten Daten
Mode = 0 oder 1, um die Art des Compactors zu wählen.
Mode 1 = Suche nach gleichen Bytes
Mode 2 = Ersetzen der Nibbles durch Codes

Nach dem Aufruf des Packers wird die Restlänge der Daten in bezug auf die Originallänge in Prozent angegeben und die Endadresse der gepackten Daten angezeigt.

Wenn man die Daten wieder entpacken will, lautet der Aufruf die Adressen

```
SYS 49265, Adr1, Adr2
```

Adr1 = Startadresse der gepackten Daten
Adr2 = Zieladresse der entpackten Daten

Die Endadresse der gepackten Daten und den Modus erkennt das Programm bei Entpacken selbst, da diese nach dem Komprimiervorgang am Anfang des Datenblocks abgelegt werden. Sie brauchen diese Angaben nicht mehr einzugeben.

Wenn Sie jetzt die Beispiele mit Hilfe des Programms

nachvollziehen, werden Sie noch einige Unterschiede zwischen unserer theoretischen Zusammenfassung und der des Computers feststellen. Laden Sie bitte dazu einen Monitor (etwa den SMON bei $9000) und füllen Sie den angegebenen Bereich $C500 mit den Daten aus dem obigen Beispiel.

Eine Kontrolle müßte ergeben:

```
M C500 C517 (Befehl zur Speicheranzeige)
C500  E0 E0 E0 E0 E0 DA 00 00
C508  00 00 00 00 1F 1F 1F 1F
C510  1F DB 00 00 00 00 00 00
```

Nun können Sie diesen Bereich einmal mit unserem Packer komprimieren. Verlassen Sie dazu den Monitor und geben Sie den folgenden Befehl ein:

```
SYS 49152, 50432, 50455, 50688, 0
```

Der Computer müßte sich nun mit den folgenden Zeilen melden:

```
ANZAHL DER GEPACKTEN DATEN IM VERHAELTNIS ZUM
ORIGINAL IN %: 70.8333333
50705 = ENDADRESSE DER GEPACKTEN DATEN READY.
```

Nun können Sie den Monitor wieder aufrufen und sich den Bereich $C600 ausgeben lassen, wo sich die komprimierten Daten befinden:

```
M C600 C610
C600  A8 0F C6 DA 04 E0 DA 00
C608  DB 05 DA 04 1F DB 00 DB
C610  05
```

Sie erkennen nun das Byte, das den Pack-Algorithmus kennzeichnet und mit $A8 über EXOR verknüpft wurde, und die beiden Bytes, die das Ende des gepackten Codes angeben. Außerdem sehen Sie, daß die Anzahl der gleichen Bytes um 1 erniedrigt angegeben wird. Dieses ist aus verschiedenen programmtechnischen Gründen sinnvoll. Bei

Code	Nibble	Nibble	Code
0	0000	0000	0
100	1111	0001	10100
10100	0001	0010	10101
10101	0010	0011	10110
10110	0011	0100	10111
10111	0100	0101	11000
11000	0101	0110	11001
11001	0110	0111	111100
11010	1000	1000	11010
11011	1001	1001	11011
11100	1010	1010	11100
11101	1100	1011	111101
111100	0111	1100	11101
111101	1011	1101	111110
111110	1101	1110	111111
111111	1110	1111	100

Tabelle 1. Jeder Code eines Nibbles bleibt eindeutig identifizierbar. Seltene Werte haben einen längeren Code.

bedingten Verzweigungen läßt sich auf diese Weise eine einfache Abfrage gestalten. Näheres können Sie in dem ausführlich kommentierten Listing nachlesen.

Soviel zu dieser Art der Packer. Wir wollen uns nun einem weiteren Compactor zuwenden, der sich auch besonders gut für Grafiken eignet. Seine Funktionsweise ist nicht ganz so trivial wie der vorangegange Algorithmus, aber auch er gehört noch zu den »einfachen« Packern. Um ihn zu verstehen, müssen wir uns nun etwas mit der Art, wie Computer Zahlen speichern, beschäftigen.

Vielleicht wissen Sie, daß der Computer nur zwei Zustände kennt: Strom an oder aus, 1 oder 0. Wenn Sie jetzt acht dieser Zustände, die man Bits nennt, zusammenfassen, erhalten Sie ein Byte. Ein Byte kann also 2 hoch 8 Zustände haben, das heißt 256 verschiedene Zahlen darstellen. Ein Byte wiederum besteht aus zwei sogenannten »Nibbles«, die aus jeweils 4 Bit bestehen. Diese Halb-Bytes oder Nibbles können nun 2 hoch 4 gleich 16 verschiedene Zahlen darstellen. Nibble-Compactoren ersetzen nun einen Nibble durch einen Code. Dabei sollte man beachten, daß die beiden Nibbles (1111) und (0000) am häufigsten in einem Byte vorkommen. Deshalb sollten deren Codes auch möglichst kurz sein. So wird nun dem Nibble mit dem Inhalt (0000) der kürzeste Code, »0«, zugeordnet. Nun muß aber jeder weitere der 15 Codes mit »1« beginnen, damit die komprimierten Daten eindeutig bleiben. Der Nibble mit (1111) erhält nun den zweitkürzesten Wert »100«. Die übrigen Nibbles werden auf weitere 14 Codes verteilt. Diese Verteilung erfolgt in Anlehnung an die des Bit-Companders, der in der 64'er-Ausgabe August 1985 veröffentlicht wurde. Wenn Sie sich einmal die Tabelle 1 ansehen, erkennen Sie, daß jeder Code eindeutig identifizierbar geblieben ist. Um dieses sicherzustellen, mußte man viele dieser neuen Codes länger als die 4 Bit, die ein Nibble lang ist, wählen. Damit sind ebenfalls die Einsatzgebiete dieses Packers charakterisiert. Speicherinhalte, die zum Beispiel aus 200 mal $BB (1011 1011) bestehen, werden deutlich länger als das Original. Wenn der Speicher aber aus vielen Null-Nibbles besteht, wird der komprimierte Datenblock dagegen bedeutend kürzer. Es ist also sinnvoll zu testen, welcher Compactor wirkungsvoller und effektiver arbeitet.

Original-$-Code	$00		$00		$00		$00	
Nibbles	0000	0000	0000	0000	0000	0000	0000	0000
Code	0	0	0	0	0	0	0	0
Bitcode	0000 0000							
Gepackter $-Code	$00							

Tabelle 2 veranschaulicht die Wirkungsweise eines Nibble-Packers am Beispiel von vier Null-Bytes

Nun wollen wir uns etwas mit der Arbeitsweise dieses Compactors beschäftigen. Sehen wir uns dazu ein Beispiel an. Folgende Bytes stehen im Speicher:

```
C500  00 00 00 00 F1 81 40 0F
C508  0F 0F
```

Wie Sie auch in dem Schema (Tabelle 1) erkennen können, zerlegt der Computer zunächst die Bytes. Aus den 4 Null-Byte erhalten wir also acht Null-Nibbles. Sie wissen bereits aus dem vorangegangenen Text, daß der Code für ein Null-Nibble eine »0« ist. Die acht Nibbles der ersten 4 Byte ergeben also nur ein neues Byte mit dem Inhalt (0000 0000), also $00. In Tabelle 2 ist dies nochmals veranschaulicht.

Bei den nun folgenden drei Codes wird die Arbeit des Compactors aber noch deutlicher. Die beiden Nibbles des ersten Bytes ergeben zusammen genau ein neues Byte. Also konnte man bei diesem Byte nicht verkürzen. Die Nibbles des sechsten Byte ($81) haben beide sogar einen um jeweils 1 Bit längeren Code. Das heißt die Codes der beiden Nibbles passen noch nicht einmal in 1 Byte. Wir haben also einen Übertrag von 2 Bit. Diese müssen in dem folgenden Byte untergebracht werden. Die beiden Übertrag-Bits und die 6 Bit der codierten Nibbles von $40 ergeben nun genau wieder ein neues Byte. Bei diesen 3 Byte war der Compactor also wirkungslos und es fand keine Verkürzung der Daten statt. Aber es wurde trotzdem codiert (Tabelle 3).

Die beiden nun folgenden Bytes kann man nun wieder zu einem Byte zusammenfassen, da sie jeweils ein Null-Nibble besitzen. Das letzte Byte liefert aber codiert nicht 8 Bit, die für ein volles Byte notwendig sind, sondern nur 4 Bit. Hierbei muß man auf eine Besonderheit hinweisen. Da-

mit man dieses »halbe« Byte ordnungsgemäß ausgeben kann, wird es mit (1111), also einem weiteren Nibble, verknüpft. Man erhält also nicht den Code $04, sonder $4F (Tabelle 4).

Dieses können Sie auch an dem Speicherauszug erkennen, in dem die codierten Byte gespeichert wurden.

```
C600  A9 0A C6 00 94 D5 2E 44
C608  4F FF
```

Original-$-Code	$F1		$81		$40	
Nibbles	1111	0001	1000	0001	0100	0000
Code	100	10100	11010	10100	10111	0
Bitcode	1001	0100	1101	0101	0010	1110
Gepackter $-Code	$94		$D5		$2E	

Tabelle 3. Nicht immer sind Nibble-Packer effektiv. Hier fand trotz Codierung keine Verkürzung statt.

Dabei enthält das erste Byte wieder das codierte Modusbyte, also das Flag für den zum Entpacken benötigten Algorithmus. Dieses ist bei diesem Packer »$A9«, im Gegensatz zu dem vorangegangenen Packer, dem der Code »$A8« zugeordnet war. Die folgenden beiden Byte enthalten, wie bereits oben erwähnt, wieder die Endadresse der komprimierten Daten.

Damit Sie nun die Künste dieses Packers ebenfalls nachvollziehen können, gebe ich Ihnen nun auch die Syntax der Befehle zum Packen beziehungsweise Entpacken an. Sie entsprechen genau denen des ersten Compactors mit einer Ausnahme. Das Modusbyte darf nun nicht Null, sondern muß Eins lauten. Sämtliche weiteren Parameter sind unverändert, auch die Adresse des SYS-Befehls. Um also Daten, die in dem Speicherbereich von $C500 bis $C57F stehen, mit dem Nibbler zu packen und die Daten dann nach $C600 zu speichern, müßten Sie folgenden Befehl eingeben:

```
SYS 49152, 50432, 50559, 50688, 1
```

Zum Entpacken müßten Sie dann, wenn die ungepackten Daten nach $C700 geschrieben werden sollen, nur den folgenden Befehl eingeben:

```
SYS 49265, 50688, 50944
```

Sollten Sie aus Versehen statt der Startadresse von den gepackten Daten eine andere Adresse eingeben, wenn der Computer also weder ein $A8 noch ein $A9 als Modus-Byte findet, gibt er eine Fehlermeldung aus. Dieses soll vor dem Entpacken eines falschen Bereiches und einem möglichen Systemabsturz schützen. Außerdem ist durch dieses Modus-Byte sichergestellt, daß die Daten nur mit dem Algorithmus entpackt werden können, mit dem sie auch verschlüsselt wurden. Sie brauchen sich also die Nummer des verwendeten Packers nicht zu merken. Es kann natürlich nicht ganz ausgeschlossen werden, daß, wenn Sie eine falsche Startadresse verwenden, aus Zufall der Computer als erstes Byte ein $A8 oder $A9 findet. Aber dieses kommt eigentlich so gut wie nicht vor.

Nun noch ein paar Informationen zu dem Programm. Es ist vollständig in Maschinensprache mit dem Assembler »Giga-Ass« (Sonderheft 21) geschrieben und verbraucht im Quellcode mehr als 15 KByte. Der Objektcode ist nicht ganz 1 KByte lang und befindet sich im Speicher von Adresse $C000 bis $C36B. Wenn Sie das Programm verschieben wollen, reicht es also aus, die Basisadresse im Quellcode (.BASE $C000) zu verändern. Das Programm ist also nicht an einen festen Speicherbereich gebunden. Achten Sie aber bitte darauf, daß sich nach dem Verschieben auch die Einsprungadressen geändert haben. Sollten Sie aber keinen Assembler haben und das Programm mit MSE eingeben, können Sie das Programm leider nicht verschieben. Um trotzdem die Flexibilität weiter zu erhöhen, ist das Programm auch fähig, Daten unter dem ROM des Computers zu lesen. Dieses ist besonders wegen der Grafik-Bilder verschiedener Zeichenprogramme notwendig, da diese ihre Bilder bei $A000 oder $E000 ablegen.

Natürlich gibt es neben diesen beiden Compactoren noch weitere, wesentlich effektivere Packer. Ihre Funktionsweise ist aber auch etwas komplizierter als die beiden besprochenen. Eine Beschreibung der Funktionsweise dieser Packer finden Sie im nächsten Artikel. Aber auch Packer für ASCII-Daten und Strings gibt es. Dabei haben einige dieser Algorithmen den Vorteil, daß sie den Speicher um einen festen Faktor zusammenfassen. Dieses liegt daran, daß man nicht 256 ASCII-Codes braucht, um alle zur Verfügung stehenden Buchstaben und Zahlen zu speichern, sondern nur 6 Bit. Diese Compactoren brauchen also nur die benutzten 6 Bit von vier Buchstaben auf 3 Byte zu 8 Bit zu verteilen. Man erhält auf diese Weise immer eine Verkürzung um 1/4 des Speicherplatzes.

(Dirk Neumeister/sk)

Original-$-Code	$0F		$0F$		$0F	
Nibbles	0000	1111	0000	1111	0000	1111
Code	0	100	0	100	0	100
Bitcode		0100	0100		0100	
Gepackter $-Code	$44				$04	

Tabelle 4. Beim dritten Wert muß mit das Nibble mit »1111« verknüpft werden, um eine ordnungsgemäße Codierung zu erreichen.

```
GIGA-ASS READY
  10; PROGRAMM ZUR DEMONSTRATION DER
  20; FUNKTIONSWEISE VON PACKERN
  30;
  40;     BY DIRK NEUMEISTER
  50;        IM  TRIESCH  28
  60;        3503  LOHFELDEN
  70;        TEL.: KS/518475
  80;
 120;
 130;
 140;
 150;
 160; PROGRAMMSTART AUF $C000 = 49152
 170;
 180          .BASE $C000
 190;
 200; DEFINITION VON ROM-ROUTINEN
 210;
 220          .EQUATE STRAUSG=$AB1E
 230          .EQUATE GETBYTE=$B79E
 240          .EQUATE PRINTFAC=$BDD7
 250          .EQUATE KOMMA=$AEFD
 260          .EQUATE INTHOLEN=$AD8A
```

Listing 1. Der Quellcode des Packprogramms im Giga-Ass-Format

```
 270          .EQUATE WANDELN=$B7F7
 280          .EQUATE FACMAL10=$BAE2
 290          .EQUATE INFAC=$B395
 300          .EQUATE COPYFACII=$BC0C
 310          .EQUATE FACDIV=$BB14
1000;
1010; ALLGEMEINER COMPACTOR START
1020;
1030; SYS 49152, ADR2, ADR3, ADR4, MODE
1040; MIT AUSGAGE DER LAENGE IN % UND DER ENDADRESSE
1050;
1060          LDA #<(REST)  ; ZEIGER FUER DEN TEXT LADEN, UM DEN
1070          LDY #>(REST)  ; HINWEIS AUF DIE RESTLAENGE AUSZUGEBEN
1080          JSR STRAUSG   ; TEXT AUSGEBEN
1090          JSR HOLEN     ; STARTADRESSE DER UNGEPACKTEN DATEN HOLEN
1100          LDA #$FB
1110          JSR COPY      ; UND DIESE NACH $FB/$FC COPIEREN
1120          JSR HOLEN     ; ENDADRESSE DER UNGEPACKTEN DATEN HOLEN
1130          INC $14       ; DIE ENDADRESSE UM EINS ERHOEHEN
1140          BNE COMP3     ;
1150          INC $15       ;
1160COMP3     LDA #$FD
1170          JSR COPY      ; UND NACH $FD/$FE COPIEREN
1180          JSR HOLEN     ; STARTADR. DER GEPACKTEN DATEN NACH $14/15 LA
DEN
1190          JSR KOMMA     ; DEN RESTLICHEN AUSDRUCK AUF KOMMA PRUEFEN
1200          JSR GETBYTE   ; EIN WEITERES BYTE IN X-REGISTER HOLEN (MODUS
)
1210          LDA #$F7      ; DIE STARTADRESSE DER GEPACKTEN DATEN
1220          JSR COPY      ; NACH $F7/$F8 COPIEREN
1230          JSR PROZENT   ; PROZENTROUTINE 1 AUFRUFEN (ERSTE BERECHNUNG)

1240          TXA           ; MODUSBYTE IN AKKU COPIEREN
1250          EOR #$A8      ; MIT EXOR 10101000 VERKNUEPFEN
1260          JSR AUSGABE   ; MODUSBYTE SPEICHERN
1270          JSR AUSGABE   ; DUMMY FUER ENDADRESSE SETZEN
1280          JSR AUSGABE   ; DUMMY FUER ENDADRESSE SETZEN
1290          TXA           ; X-REGISTER COPIEREN, FLAGS WERDEN GESETZT
1300          BNE COMP1     ; WENN <> 0, DANN MIT NIBBLER ARBEITEN
1310          JSR PACKEN    ; SONST MIT PACKER ARBEITEN
1320          JMP COMP2
1330COMP1     JSR NIBBLER   ; NIBBLER AUFRUFEN
1340          LDA #$EE      ; AKKU MIT $EE LADEN UND NIBBLER ERNEUT AUFRUF
EN
1350          JSR NIBBLER3  ; DAMIT DAS LETZTE BYTE AUCH GESPEICHERT WIRD
1360COMP2     LDY #$01
1370          LDA $F7       ; ENDADRESSE SPEICHERN, WO VORHER DIE DUMMIES
1380          STA ($61),Y   ; GESTANDEN HABEN
1390          INY
1400          LDA $F8
1410          STA ($61),Y
1420          JSR PROZENT2  ; PROZENTROUTINE 2 AUFRUFEN
1430          JSR PRINTFAC  ; RESTLAENGE DES PROGRAMMS AUSGEBEN
1440          JSR $AAD7     ; 2 MAL RETURN AUSGEBEN
1450          JSR $AAD7
1460          LDA $F8       ; ENDADRESSE DER GEPACKTEN DATEN LADEN
1470          LDX $F7
1480          JSR $BDCD     ; UND EBENFALLS AUSGEBEN
1490          LDA #<(ENDADR); ZEIGER FUER DEN TEXT DER ENDADRESSE LADEN
1500          LDY #>(ENDADR)
1510          JSR STRAUSG   ; UND DEN TEXT AUSGEBEN
1520          RTS
2000;
2010; ALLGEMEINER DECOMPACTOR START
2020; SYS 49265, ADR2, ADR3
2030;
2040DECOMP    JSR HOLEN     ; STARTADRESSE DER GEPACKTEN DATEN HOLEN
2050          LDA #$FB
2060          JSR COPY      ; UND NACH $FB/$FC COPIEREN
2070          JSR NEXTBYTE  ; MODUSBYTE HOLEN
2080          PHA           ; UND AUF STAPEL LEGEN
2090          JSR NEXTBYTE  ; ENDADRESSE DER GEPACKTEN DATEN AUS DEM
2100          STA $35       ; SPEICHER LESEN UND NACH $35/$36 SPEICHERN
2110          JSR NEXTBYTE
2120          STA $36
2130          JSR HOLEN     ; STARTADRESSE DER UNGEPACKTEN DATEN HOLEN
2140          LDA #$F7      ;
2150          JSR COPY      ; NACH $F7/$F8 COPIEREN
2160          PLA           ; STAPEL LESEN UND IN AKKU SPEICHERN
2170          EOR #$A8      ; CODIERUNG AUFHEBEN
2180          BNE DECOMP1   ; WENN AKKU <> 0, DANN NIBBLER AUFRUFEN
2190          JSR ENTPACKEN ; SONST PACKER AUFRUFEN
2200          RTS
2210DECOMP1   TAX           ; AKKU IN X-REGISTER COPIEREN
2220          DEX           ; X-REGISTER UM EINS VERRINGERN
2230          BNE DECOMP2   ; WENN X <> 0 (FEHLER), FALSCHE STARTADRESSE
2240          JSR DENI      ; SONST NIBBLER AUFRUFEN
2250          RTS
2260DECOMP2   LDA #<(FEHLER); AUSGABE EINER FEHLERMELDUNG, DA DAS DECODIER
TE
2270          LDY #>(FEHLER); MODUSBYTE WEDER NULL NOCH EINS WAR ==>
2280          JSR STRAUSG   ; FALSCHE STARTADRESSE ANGEGEBEN
2290          RTS
3000;
3010; UNTERROUTINEN FUER DIE
3020; ALLGEMEINEN EINSPRUNGADRESSEN
3030;
3040HOLEN     JSR KOMMA     ; PRUEFT AUF KOMMA
3050          JSR INTHOLEN  ; 16-BIT ZAHL IN FAC HOLEN
3060          JSR WANDELN   ; FAC WANDELN IN INTEGER $14/$15
3070          RTS           ; ZURUECK
3080;
3090COPY      LDY #$01      ; ZAHLEN DER SPEICHERSTELLEN $14/$15
3100          STA COPY1+1   ; IN DIE SPEICHERSTELLE COPIEREN, DIE IM
3110COPY2     LDA $14,Y     ; AKKU (+1) STEHT
3120COPY1     STA $00,Y
3130          DEY
3140          BEQ COPY2
3150          RTS           ; UND WIEDER ZURUECK
4000;
4010;  PACKER
4020;  $FB/$FC = BEGINN DER UNGEPACKTEN DATEN
4030;  $FD/$FE = ENDE DER UNGEPACKTEN DATEN
4040;  $F7/$F8 = BEGINN DER GEPACKTEN DATEN
4050;
4060PACKEN    JSR NEXTBYTE  ; NAECHSTES BYTE IN AKKU HOLEN
4070          TAX           ; BYTE VOM AKKU NACH X KOPIEREN
4080          JSR VERGLEICHE; BYTE IM AKKU MIT FOLGENDEN BYTES VERGLEICHEN

4090          STY $A5       ; ANZAHL-1 DER GLEICHEN BYTES IN Y UND $A5
4100          CPY #03       ; ANZAHL-1 DER BYTES MIT #03 VERGLEICHEN
4110          BCC PACKEN1   ; WENN ANZAHL-1 < 3, DANN NICHT PACKEN
4120          TYA           ; ANZAHL-1 IN DEN AKKU KOPIEREN
4130          CLC           ; ADDITION VORBEREITEN
4140          ADC $FB       ; UM ANZAHL-1 DER GLEICHEN BYTES DEN
4150          STA $FB       ; ZEIGER FUER UNGEPACKTE DATEN NACH VORNE, VER
-
4160          BCC PACKEN2   ; SCHIEBEN, DAMIT NICHT DOPPELT ZUSAMMEN-
4170          INC $FC       ; GEFASST WIRD
4180PACKEN2   CPX #00       ; VERGLEICHEN DER BYTES MIT $00
4190          BEQ PACKEN3   ; WENN JA, DANN MIT $DB CODIEREN
4200          LDA #$DA      ; SONST MIT $DA BEGINNEN
4210          JSR AUSGABE   ; $DA SPEICHERN
4220          LDA $A5       ; ANZAHL-1 DER GLEICHEN BYTES IN AKKU
4230          JSR AUSGABE   ; UND ABSPEICHERN
4240          TXA           ; CODE DER GLEICHEN BYTES IN AKKU
4250          JSR AUSGABE   ; UND EBENFALLS SPEICHERN
4260          BNE PACKEN4   ; UNBEDINGTER SPRUNG, DA AKKU = #$FF
4270PACKEN3   LDA #$DB      ; #$DB ALS CODE FUER 00 IN AKKU LADEN
4280          JSR AUSGABE   ; UND IM SPEICHER ABLEGEN
4290          LDA $A5       ; ANZAHL-1 DER $00-BYTES IN AKKU
4300          JSR AUSGABE   ; UND EBENFALLS SPEICHERN
4310          BNE PACKEN4   ; UNBEDINGTER SPRUNG
4320PACKEN1   CPX #$DA      ; VERGLEICHEN OB BYTE ZUFAELLIG $DA
4330          BEQ PACKEN5
4340          CPX #$DB      ; ODER $DB IST
4350          BNE PACKEN6   ; WENN NICHT, VERZWEIGEN (EINFACHE AUSG.)
4360PACKEN5   TXA           ; $DA ODER $DB VOM X-REGISTER IN AKKU COPIEREN

4370          JSR AUSGABE   ; UND AUSGEBEN
4380          LDA #00       ; ANZAHL-1 LADEN
4390          JSR AUSGABE   ; UND EBENFALLS ABSPEICHERN
4400          BNE PACKEN4   ; UNBEDINGTER SPRUNG
4410PACKEN6   TXA           ; EINFACHE AUSGABE, D.H. CODE IN AKKU
4420          JSR AUSGABE   ; UND IM SPEICHERN ABLEGEN
4430PACKEN4   LDA $FB
```

```
4440            LDX $FC      ; VERGLEICHEN OB ENDE, WENN NEIN, DANN
4450            CPX $FE
4460            BNE PACKEN7
4470            CMP $FD
4480PACKEN7     BCC PACKEN   ; VON VORNE WIEDER BEGINNEN
4490            RTS          ; SONST RUECKSPRUNG AUS DER UNTERROUTINE
5000;
5010;  ENTPACKEN
5020;  $F7/$F8 = BEGINN DER UNGEPACKTEN DATEN
5030;  $FB/$FC = BEGINN DER GEPACKTEN DATEN
5040;  $35/$36 = ENDE DER GEPACKTEN DATEN
5050;
5060ENTPACKEN JSR NEXTBYTE ; NAECHSTES BYTE IN DEN AKKU LADEN
5070            CMP #$DB     ; MIT DEM ERKENNUNGSCODE $DB VERGLEICHEN
5080            BNE ENTPACKEN1; WENN AKKU <> #$DB DANN VERZWEIGEN
5090            JSR NEXTBYTE ; SONST NAECHSTES BYTE HOLEN
5100            CMP #00      ; TEST, OB ANZAHL-1 DER GLEICHEN BYTE > 0
5110            BNE ENTPACKEN2; WENN NEIN, DANN VERZWEIGE
5120            LDA #$DB     ; ES WAR ALSO IM ORIGINAL EIN $DB, AKKU=$DB
5130            BNE ENTPACKEN3; UND VERZWEIGEN UM DIREKT SPEICHERN
5140ENTPACKEN2 TAX           ; ANZAHL-1 DER$00 IN D. X-REGISTER COPIEREN
5150            LDA #00      ; AKKU MIT DEM CODE (HIER #$00) LADEN
5160            JMP ENTPACKEN8; UND ENTPACKEN
5170ENTPACKEN1 CMP #$DA      ; VERGLEICHEN OB BYTE = ERKENNUNGSCODE $DA IST

5180            BNE ENTPACKEN3; WENN NEIN, VERZWEIGEN UND BYTE AUSGEBEN
5190            JSR NEXTBYTE ; NAECHSTES BYTE HOLEN
5200            CMP #00      ; TEST, CODE FUER GEPACKT ODER FUER $DA
5210            BNE ENTPACKEN4; WENN CODE FUER GEPACKT, DANN VERZWEIGE
5220            LDA #$DA     ; SONST LADE AKKU MIT $DA
5230            BNE ENTPACKEN3; UND VERZWEIGE ZUR DIREKTEN AUSGABE
5240ENTPACKEN4 TAX           ; ZAHL DER GLEICHEN BYTE -1 IN X-REIGSTER
5250            JSR NEXTBYTE ; NOCH EIN BYTE HOLEN, (CODE DER GL. BYTE)
5260ENTPACKEN8 INX           ; X MAL DEN INHALT DES AKKUS SPEICHERN
5270            LDY #00
5280ENTPACKEN6 STA ($F7),Y
5290            INY
5300            DEX
5310            BNE ENTPACKEN6; NOCH NICHT FERTIG, DANN WEITERMACHEN
5320            DEY
5330            TYA
5340            SEC
5350            ADC $F7      ; SPEICHERZEIGER WEITERSTELLEN, DAMIT
5360            STA $F7      ; MAN NICHT DOPPELT IN DEN SPEICHER
5370            BCC ENTPACKEN9; GESCHRIEBEN WIRD
5380            INC $F8
5390            BEQ ENTPACKEN9
5400ENTPACKEN3 JSR AUSGABE  ; BYTE IN DEN SPEICHER SCHREIBEN
5410ENTPACKEN9 LDA $35      ; TEST, OB SCHON ALLE BYTES ENTPACKT WURDEN
5420            LDX $36
5430            CPX $FC
5440            BNE ENTPACKEN ; WENN NEIN, WEITERMACHEN
5450            CMP $FB
5460            BNE ENTPACKEN
5470            RTS          ; SONST FERTIG, D.H. RUECKSPRUNG
6000;
6010; UNTERROUTINEN FUER DEN PACKER
6020; UND DEN NIBBLER
6030;
6040;
6050;NAECHSTES BYTE HOLEN
6060;
6070NEXTBYTE  SEI
6080            LDY #$34     ; ROM AUSSCHALTEN
6090            STY $01
6100            LDY #00      ; Y=NULL SETZEN
6110            LDA ($FB),Y  ; DEN AKKU MIT DEM WERT DER SPEICHERSTELLE LAD
EN
6120            LDY #$37     ; ROM WIEDER ANSCHALTEN
6130            STY $01
6140            CLI
6150            INC $FB      ; ZEIGER FUER BYTE HOLEN $(FB/FC) UM
6160            BNE NEXTBEND ; EINS ERHOEHEN
6170            INC $FC
6180NEXTBEND  RTS            ; UND WIEDER ZURUECKSPRINGEN
6190;
6200;AUSGABE
6210;
6220AUSGABE   LDY #00        ; LADE Y MIT NULL
6230            STA ($F7),Y  ; SPEICHERE AKKU IN DIE SPEICHERSTELLE,
6240            INC $F7      ; AUF DIE DER ZEIGER $(F7/F8) ZEIGT
6250            BNE AUSGABEND ; ERHOEHE DEN ZEIGER DANN UM EINS
6260            INC $F8
6270AUSGABEND LDA #$FF       ; LADE AKKU=$FF (FLAGS WERDEN BEEINFLUSST)
6280            RTS          ; VERLASSE DIESE UNTERROUTINE WIEDER
6290;
6300;VERLGEICHEN
6310;
6320VERGLEICHE LDY #00       ; HIER WIRD DAS Y-REGISTER NULL GESETZT
6330VERGL1     CMP ($FB),Y   ; VERGLEICHE DEN INHALT DES AKKUS MIT DER
6340            BNE VERGLEND ; SPEICHERSTELLE, AUF DIE DER ZEIGER ZEIGT
6350            INY          ; ZAEHLE, WIE OFT SICH DIE BYTES GLEICHEN
6360            BNE VERGL1   ; MAX. ANZAHL IST DABEI 255 (FF)
6370            DEY
6380VERGLEND  RTS            ; RUECKSPRUNG WENN DIE BYTES VERSCHIEDEN
6390            NOP          ; SIND ODER DIE ANZAHL GROESSER ALS 255 IST
6400;
6410; NIBBLER ZUSAMMENFASSEN
6420;
6430ZUSAM     TAY            ; WERT DES NIBBLES IN DAS Y-REGISTER KOPIEREN
6440            LDA BITLAENGE,Y; DIE BITLAENGE DES NEUEN CODES IN DEN AKKU
6450            TAX          ; LADEN UND DANN NACH X KOPIEREN
6460            LDA CODE,Y   ; DEN NIBBLECODE AUFGRUND SEINES WERTES LADEN
6470            LDY $F9      ; FREIE BITS IN DAS Y-REGISTER LADEN
6480ZUSAM2    ASL
6490            ROL $FA      ; BITS AUSTAUSCHEN
6500            DEY
6510            BNE ZUSAM1   ; WENN BYTE NOCH NICHT VOLL, DANN VERZWEIGE
6520            PHA          ; AKKU AUF DEN STAPEL LEGEN
6530            LDA $FA      ; DEN NEU ZUSAMMENGESTELLTEN WERT LADEN
6540            JSR AUSGABE  ; UND AUSGEBEN
6550            LDY #$08     ; ES SIND WIEDER ACHT BITS FREI
6560            PLA          ; AKKU WIEDER VOM STAPEL HOLEN
6570ZUSAM1    DEX            ; NOCH VERBLEIBENDE CODELAENGE UM EINS VERMIND
ERN
6580            BNE ZUSAM2   ; UND WEITERMACHEN, BIS DER CODE COPIERT WURDE

6590            STY $F9      ; ZAHL DER NOCH FREIEN BITS ZWISCHENSPEICHERN
6600            RTS          ; UND ZURUECKSPRINGEN
6610;
6620; NIBBLER WIEDER AUSEINANDER ZIEHEN
6630;
6640ENTNI     LDX #$0F       ; SCHLEIFENZAEHLER (X-REGISTER) MIT 16 LADEN
6650ENTNI2    LDA $FD
6660            AND MASKE,X  ; DAS BYTE MIT DER MASKE VERKNUEPFEN
6670            CMP CODE,X   ; ZAHL DER NOCH FREIEN BYTES UM EINS VERRINGER
N
6680            BEQ ENTNI1   ; WENN UEBEREINSTIMMUNG, DANN VERZWEIGEN
6690            DEX          ; SONST ZAEHLER UM EINS ERNIEDRIGEN
6700            BNE ENTNI2   ; UND WEITER SUCHEN
6710ENTNI1    STX $FA        ; CODENUMMER ZWISCHENSPEICHERN
6720            LDA BITLAENGE,X; BITLAENGE LADEN
6730            TAX          ; UND IN DAS X-REGISTER KOPIEREN
6740            LDY $F9      ; ANZAHL DER RESTLICHEN BITS IN Y LADEN
6750ENTNI4    ASL $FE        ; UND AUSTAUSCHEN
6760            ROL $FD
6770            DEY          ; ANZAHL DER BITS, DIE NOCH ZU VERFUEGUNG STEH
EN
6780            BNE ENTNI3   ; WENN DIESE GLEICH NULL IST, NICHT VERZWEIGEN

6790            TYA          ;
6800            PHA          ; Y-REGISTER UND AKKU AUF STAPEL RETTEN
6810            TXA
6820            PHA
6830            JSR ENDE     ; TEST, OB BEREITS ALLE BYTES DECOMPREMIERT
6840            CPY #$F8     ; WURDEN, WENN JA, IST Y = $F8 UND
6850            BEQ ENTNI6   ; DANN VERZWEIGEN
6860            JSR NEXTBYTE ; SONST NEUES BYTE HOLEN
6870            STA $FE      ; UND AUF $FE SPEICHERN
6880ENTNI6    PLA            ; AKKU UND X-REGISTER WIEDER VOM STAPEL HOLEN
6890            TAX
6900            PLA
6910            CPY #$F8     ; WENN ALLE BYTES DECODIERT WURDEN, (Y=$F8)
6920            BEQ ENTNI5   ; DANN VERZWEIGEN ZUM RUECKSPRUNG
```

Listing 1. Quellcode des Packers (Fortsetzung)

```
6930          LDY #$08      ; ES SIND WIEDER 8 BITS DA
6940ENTNI3    DEX           ; BITLAENGE DER CODE UM EINS ERNIEDRIGEN
6950          BNE ENTNI4    ; WENN NOCH NICHT NULL, WIEDER VON VORNE
6960          STY $F9       ; ANZAHL DER NOCH FREIEN BITS SPEICHERN
6970          LDA $FA       ; CODEWERT LADEN (DIES IST EIN NIBBLE)
6980ENTNI5    RTS           ; UND ZURUECK
6990;
7000; TEST AUF ENDE
7010;
7020ENDE      LDA $35       ; ENDADRESSE IN AKKU UND X-REGISTER LADEN
7030          LDX $36
7040          CPX $FC       ; MIT DER AKTUELLEN LADEADRESSE VERGLEICHEN
7050          BNE ENDE1
7060          CMP $FB
7070          BNE ENDE1     ; WENN KLEINER, DANN WEITER
7080          LDY #$F8      ; WENN GLEICH, Y-CODE FUER ENDE ($F8) LADEN
7090ENDE1     RTS           ; UND ZURUECK SPRINGEN
8000;
8010;  NIBBLE - PACKER
8020;  $FB/$FC = BEGINN DER UNGEPACKTEN DATEN
8030;  $FD/$FE = ENDE DER UNGEPACKTEN DATEN
8040;  $F7/$F8 = BEGINN DER GEPACKTEN DATEN
8050;
8060NIBBLER   LDA #$08      ; ANZAHL DER FREIEN BITS IN EINEM BYTE LADEN
8070          STA $F9       ; UND NACH $F9 SPEICHERN
8080NIBBLER1  JSR NEXTBYTE  ; NEUES BYTE HOLEN
8090NIBBLER3  PHA           ; BYTE AUF STAPEL ZWISCHENSPEICHERN
8100          LSR           ; HI-NIBBLE NACH RECHTS SCHIEBEN
8110          LSR           ; LOW-NIBBLE LOESCHEN
8120          LSR
8130          LSR
8140          JSR ZUSAM     ; UND CODIEREN
8150          PLA           ; BYTE WIEDER VOM STAPEL HOLEN
8160          AND #$0F      ; HIGH-NIBBLE LOESCHEN, LOW NIBBLE BLEIBT UEBR
IG
8170          JSR ZUSAM     ; UND AUCH DIESES CODIEREN
8180          LDA $FB
8190          LDX $FC       ; TESTEN, OB BEREITS ALLE BYTES CODIERT WURDEN

8200          CPX $FE
8210          BNE NIBBLER2  ; WENN NEIN, WEITERMACHEN
8220          CMP $FD
8230NIBBLER2  BCC NIBBLER1
8240          RTS           ; SONST RUECKSPRUNG
9000;
9010;  ENTPACKEN
9020;  $F7/$F8 = BEGINN DER UNGEPACKTEN DATEN
9030;  $FB/$FC = BEGINN DER GEPACKTEN DATEN
9040;  $35/$36 = ENDE DER GEPACKTEN DATEN
9050;
9060DENI      INC $35       ; ENDADRESSE UM EINS ERHOEHEN
9070          BNE DENI3
9080          INC $36
9090DENI3     JSR NEXTBYTE  ; EIN BYTE HOLEN
9100          STA $FD       ; UND ZWISCHENSPEICHERN
9110          JSR NEXTBYTE  ; EIN WEITERES BYTE HOLEN
9120          STA $FE       ; EBENFALLS ZWISCHENSPEICHERN
9130          LDA #$08      ; ANZAHL DER FREIEN BITS IN EINEM BYTE IST 08
9140          STA $F9       ; DIESE EBENFALLS SPEICHERN
9150DENI1     JSR ENTNI     ; HI-NIBBLE DECODIEREN
9160          CPY #$F8      ; TEST AUF ENDE (Y = $F8)
9170          BEQ DENI2     ; WENN JA, VERZWEIGEN ZUM RUECKSPRUNG
9180          ASL           ; SONST HI-NIBBLE NACH LINKS VERSCHIEBEN
9190          ASL
9200          ASL
9210          ASL
9220          PHA           ; AUF STAPEL LEGEN
9230          JSR ENTNI     ; LOW-NIBBLE DECODIEREN
9240          PLA           ; AKKU WIEDER VOM STAPEL LADEN
9250          CPY #$F8      ; WENN ENDE DER DATEN (Y = $F8)
9260          BEQ DENI2     ; DANN VERZWEIGE ZUM RUECKSPRUNG
9270          ORA $FA       ; LOW- UND HIGH-NIBBLE VERKNUEPFEN
9280          JSR AUSGABE   ; UND AUSGEBEN
9290          JMP DENI1     ; WEITERMACHEN
9300DENI2     RTS           ; RUECKSPRUNG
10000;
10010; PROZENTSATZ DER COMPREMIERUNG
10020; BERECHNEN   (ERSTER AUFRUF)
10030; $FB/$FC = BEGINN DER UNGEPACKTEN DATEN
10040; $FD/$FE = ENDE DER UNGEPACKTEN DATEN
10050; ==> $35/$36 = ERGEBNIS DER DIFFERENZ
10060; $F7/$F8 (BEGINN DER GEPACKTEN DATEN) IN $61/$62 ZWISCHENSPEICHERN
10070;
10080PROZENT  CLD           ;SUBTRAKTION VORBEREITEN
10090         SEC
10100         LDA $FD       ; LAENGE DER UNGEPACKTEN
10110         SBC $FB       ; DATEN BERECHNEN UND IN
10120         STA $35       ; $35/$36 ZWISCHENSPEICHERN
10130         LDA $FE
10140         SBC $FC
10150         STA $36
10160         LDA $F7       ; BEGINN DER GEPACKTEN DATEN RETTEN
10170         STA $61       ; INDEM MAN SIE IN $61/$62 ZWISCHENSPEICHERT
10180         LDA $F8
10190         STA $62
10200         RTS           ; UND ZURUECKSPRINGEN
10210;
10220; PROZENTSATZ DER COMPREMIERUNG
10230; BERECHNEN   (ZWEITER AUFRUF)
10240; $F7/$F8 = ENDE DER GEPACKTEN DATEN
10250; $61/$62 = BEGINN DER GEPACKTEN DATEN
10260; $35/$36 = LAENGE DER UNGEPACKTEN DATEN
10270; BEIM RUECKSPRUNG ENTHAELT FAC I DEN PROZENTANTEIL
10280;
10290PROZENT2 CLD           ; SUBTRAKTION VORBEREITEN
10300         SEC
10310         LDA $F7       ; LAENGE DER GEPACKTEN DATEN BERECHNEN
10320         SBC $61
10330         TAY
10340         LDA $F8
10350         SBC $62
10360         JSR INFAC     ; UND IN FLIESSKOMMAAKKU I (FAC I) WANDELN
10370         JSR COPYFACII ; UND IN DEN FAC II COPIEREN
10380         LDY $35
10390         LDA $36       ; LAENGE DER UNGEPACKTEN DATEN LADEN
10400         JSR INFAC     ; UND EBENFALLS IN FAC SPEICHERN
10410         JSR FACDIV    ; DIVISION VON FAC UND FAC II
10420         JSR FACMAL10  ; FAC = FAC *10
10430         JSR FACMAL10  ; UND NOCHMAL (==> PROZENTANTEIL AUSRECHNEN)
10440         RTS           ; UND WIEDER ZURUECK
11000;
11010; DATEN, TEXT UND FEHLERMELDUNGEN
11020; FUER DAS PROGRAMM
11030;
11040CODE     .BYTE $00,$A0,$A8; NEUE CODES, DIE STATT EINES NIBBLES
11050         .BYTE $B0,$B8,$C0; GESPEICHERT WERDEN
11060         .BYTE $C8,$F0,$D0
11070         .BYTE $D8,$E0,$F4
11080         .BYTE $E8,$F8,$FC
11090         .BYTE $80
11100MASKE    .BYTE $80,$F8,$F8; MASKE, UM DIE UNNOETIGEN BITS ZU LOESCHEN

11110         .BYTE $F8,$F8,$F8; UND DIE CODES ZU IDENTIFIZIEREN
11120         .BYTE $F8,$FC,$F8
11130         .BYTE $F8,$F8,$FC
11140         .BYTE $F8,$FC,$FC
11150         .BYTE $E0
11160BITLAENGE .BYTE $01,$05,$05; LAENGE DER NEUEN CODES
11170         .BYTE $05,$05,$05
11180         .BYTE $05,$06,$05
11190         .BYTE $05,$05,$06
11200         .BYTE $05,$06,$06
11210         .BYTE $03
11220REST     .BYTE 13
11230         .TEXT "ANZAHL DER GEPACKTEN DATEN IM VERHAELT-
11240         .BYTE 13
11250         .TEXT "NIS ZUM ORIGINAL IN %:
11260         .BYTE 0
11270ENDADR   .TEXT " = ENDADRESSE DER GEPACKTEN DATEN
11280         .BYTE 0
11290FEHLER   .TEXT "WIE BITTE? DIE STARTADRESSE IST FALSCH!
11300         .BYTE 13,13,0
```

Listing 1. Quellcode des Packers (Schluß)

```
Name : packer.obj              c000 c36e
-----------------------------------------
c000 : a9 c4 a0 c2 20 1e ab 20   6e
c008 : a8 c0 a9 fb 20 b2 c0 20   d5
c010 : a8 c0 e6 14 d0 02 e6 15   37
c018 : a9 fd 20 b2 c0 20 a8 c0   4f
c020 : 20 fd ae 20 9e b7 a9 f7   2d
c028 : 20 b2 c0 20 5b c2 8a 49   5e
c030 : a8 20 82 c1 20 82 c1 20   1e
c038 : 82 c1 8a d0 06 20 c1 c0   41
c040 : 4c 4b c0 20 08 c2 a9 ee   81
c048 : 20 0f c2 a0 01 a5 f7 91   f5
c050 : 61 c8 a5 f8 91 61 20 72   27
c058 : c2 20 d7 bd 20 d7 aa 20   84
c060 : d7 aa a5 f8 a6 f7 20 cd   5b
c068 : bd a9 21 a0 c3 20 1e ab   63
c070 : 60 20 a8 c0 a9 fb 20 b2   83
c078 : c0 20 6d c1 48 20 6d c1   9b
c080 : 85 35 20 6d c1 85 36 20   b7
c088 : a8 c0 a9 f7 20 b2 c0 68   65
c090 : 49 a8 d0 04 20 1c c1 60   8d
c098 : aa ca d0 04 20 2a c2 60   7b
c0a0 : a9 43 a0 c3 20 1e ab 60   ee
c0a8 : 20 fd ae 20 8a ad 20 f7   fd
c0b0 : b7 60 a0 01 8d bb c0 b9   0d
c0b8 : 14 00 99 f7 00 88 f0 f7   2a
c0c0 : 60 20 6d c1 aa 20 8f c1   31
c0c8 : 84 a5 c0 03 90 2a 98 18   9c
c0d0 : 65 fb 85 fb 90 02 e6 fc   c2
c0d8 : e0 00 f0 10 a9 da 20 82   ed
c0e0 : c1 a5 a5 20 82 c1 8a 20   82
c0e8 : 82 c1 d0 23 a9 db 20 82   e2
c0f0 : c1 a5 a5 20 82 c1 d0 17   99
c0f8 : e0 da f0 04 e0 db d0 0b   48
c100 : 8a 20 82 c1 a9 00 20 82   93
c108 : c1 d0 04 8a 20 82 c1 a5   ec
c110 : fb a6 fc e4 fe d0 02 c5   44
c118 : fd 90 a6 60 20 6d c1 c9   1b
c120 : db d0 11 20 6d c1 c9 00   b8
c128 : d0 04 a9 db d0 2f aa a9   65
c130 : 00 4c 47 c1 c9 da d0 25   61
c138 : 20 6d c1 c9 00 d0 04 a9   a2
c140 : da d0 1a aa 20 6d c1 e8   a4
c148 : a0 00 91 f7 c8 ca d0 fa   68
c150 : 88 98 38 65 f7 85 f7 90   8c
c158 : 07 e6 f8 f0 03 20 82 c1   ed
c160 : a5 35 a6 36 e4 fc d0 b4   f3
c168 : c5 fb d0 b0 60 78 a0 34   2a
c170 : 84 01 a0 00 b1 fb a0 37   89
c178 : 84 01 58 e6 fb d0 02 e6   8c
c180 : fc 60 a0 00 91 f7 e6 f7   39
c188 : d0 02 e6 f8 a9 ff 60 a0   8f
c190 : 00 d1 fb d0 04 c8 d0 f9   4f
c198 : 88 60 ea a8 b9 b4 c2 aa   c2
c1a0 : b9 94 c2 a4 f9 0a 26 fa   67
c1a8 : 88 d0 09 48 a5 fa 20 82   9b
c1b0 : c1 a0 08 68 ca d0 ee 84   c8
c1b8 : f9 60 a2 0f a5 fd 3d a4   f4
c1c0 : c2 dd 94 c2 f0 03 ca d0   e2
c1c8 : f3 86 fa bd b4 c2 aa a4   ca
c1d0 : f9 06 fe 26 fd 88 d0 19   ea
c1d8 : 98 48 8a 48 20 f9 c1 c0   9a
c1e0 : f8 f0 05 20 6d c1 85 fe   8f
c1e8 : 68 aa 68 c0 f8 f0 09 a0   54
c1f0 : 08 ca d0 dd 84 f9 a5 fa   f2
c1f8 : 60 a5 35 a6 36 e4 fc d0   6d
c200 : 06 c5 fb d0 02 a0 f8 60   cc
c208 : a9 08 85 f9 20 6d c1 48   5b
c210 : 4a 4a 4a 4a 20 9b c1 68   12
c218 : 29 0f 20 9b c1 a5 fb a6   cb
c220 : fc e4 fe d0 02 c5 fd 90   cf
c228 : e3 60 e6 35 d0 02 e6 36   c1
c230 : 20 6d c1 85 fd 20 6d c1   42
c238 : 85 fe a9 08 85 f9 20 ba   c6
c240 : c1 c0 f8 f0 15 0a 0a 0a   9b
c248 : 0a 48 20 ba c1 68 c0 f8   2a
c250 : f0 08 05 fa 20 82 c1 4c   9b
c258 : 3e c2 60 d8 38 a5 fd e5   9f
c260 : fb 85 35 a5 fe e5 fc 85   3e
c268 : 36 a5 f7 85 61 a5 f8 85   52
c270 : 62 60 d8 38 a5 f7 e5 61   b4
c278 : a8 a5 f8 e5 62 20 95 b3   d2
c280 : 20 0c bc a4 35 a5 36 20   04
c288 : 95 b3 20 14 bb 20 e2 ba   3f
c290 : 20 e2 ba 60 00 a0 a8 b0   e5
c298 : b8 c0 c8 f0 d0 d8 e0 f4   42
c2a0 : e8 f8 fc 80 80 f8 f8 f8   f9
c2a8 : f8 f8 f8 fc f8 f8 f8 fc   2f
c2b0 : f8 fc fc e0 01 05 05 05   d8
c2b8 : 05 05 05 06 05 05 05 06   db
c2c0 : 05 06 06 03 0d 41 4e 5a   73
c2c8 : 41 48 4c 20 44 45 52 20   3c
c2d0 : 47 45 50 41 43 4b 54 45   61
c2d8 : 4e 20 44 41 54 45 4e 20   59
c2e0 : 49 4d 20 56 45 52 48 41   2d
c2e8 : 45 4c 54 2d 0d 4e 49 53   1d
c2f0 : 20 5a 55 4d 20 4f 52 49   95
c2f8 : 47 49 4e 41 4c 20 49 4e   27
c300 : 20 25 3a 20 20 20 20 20   09
c308 : 20 20 20 20 20 20 20 20   08
c310 : 20 20 9d 9d 9d 9d 9d 9d   d4
c318 : 9d 9d 9d 9d 9d 9d 9d 9d   17
c320 : 00 20 3d 20 45 4e 44 41   de
c328 : 44 52 45 53 53 45 20 44   ba
c330 : 45 52 20 47 45 50 41 43   f2
c338 : 4b 54 45 4e 20 44 41 54   9a
c340 : 45 4e 00 57 49 45 20 42   5b
c348 : 49 54 54 45 3f 20 44 49   12
c350 : 45 20 53 54 41 52 54 41   7f
c358 : 44 52 45 53 53 45 20 49   f4
c360 : 53 54 20 46 41 4c 53 43   f9
c368 : 48 21 0d 0d 00 00 ff 00   26
```

Listing 2. »PACKER.OBJ«
Die lauffähige Packroutine. Bitte mit dem MSE (Seite 159) eingeben.

Von Packern und Kompressoren (2)

Im zweiten Teil unseres Grundlagenartikels wollen wir uns mit einem Packverfahren auseinandersetzen, das zu einem der ausgefeiltesten zählt: dem Lempel-Ziv-Verfahren.

Zunächst folgende Überlegung: mit 8 Bit kann man bekanntlich 256 verschiedene Kombinationen darstellen – eben die 256 Byte. Mit 9 Bit hingegen sind es schon 512 Kombinationen, mit 10 Bit 1024, mit 11 Bit 2048 etc. Könnte man die 512 9-Bit-Werte nicht ausnutzen um damit sowohl die 256 Byte als auch 256 besonders häufige 2-Byte-Folgen zu je 9 Bit zu codieren? Man würde dabei manchmal 7 Bit gewinnen (bei 2-Byte-Folgen), oft jedoch 1 Bit verlieren (bei »einzelnen« Bytes). Ob sich das am Ende lohnt, sei dahingestellt. Vielleicht wäre es günstiger, alle Bytes zu 10 Bit zu codieren und die verbleibenden 1024–256 = 768 Codierungen für die häufigsten 2-Byte-Sequenzen zu reservieren. Dabei würde häufiger als im zuletzt skizzierten »9-Bit-Fall« etwas eingespart – und zwar jetzt je 6 Bit – aber ziemlich oft würden 2 Bit bei einzelnen zu codierenden Bytes eingebüßt. Dieses Beispiel ließe sich sicher beliebig fortsetzen, nähme man statt 9- oder 10-Bit-Codes beispielsweise sogar 11-, 12- oder 13-Bit-Codes. Auch könnte man statt der oben empfohlenen 2-Byte-Sequenzen längere nehmen. Zwar kommen 3-Byte-Sequenzen deutlich seltener vor als solche zu 2 Byte, jedoch wird durch eine Codierung mit zum Beispiel 9 oder 10 Bit offensichtlich auch einiges mehr als bei 2-Byte-Folgen eingespart.

Offenbar hat ein solcher Packer, entscheidet man sich für eines der im letzten Absatz skizzierten Verfahren, zumindest drei Nachteile:

1) Hat man sich für eine Codierung zu je 10 Bit entschlossen – vielleicht, weil dies für das zu packende Programm ein besonders gutes Ergebnis bringt – so muß man den ganzen Kompressionsvorgang hindurch bei diesen 10 Bit bleiben, auch, wenn es vielleicht sinnvoll wäre, einen Teil des Programms zu je 9 Bit zu codieren. Die Codewortlänge ist also statisch und läßt sich demnach nicht flexibel auf das zu packende Programm einstellen.

2) Ähnliches gilt für die Sequenzlängen. Entweder, man verkürzt nur 2-Byte-Sequenzen oder nur 3-Byte-Sequenzen. Es ist zwar prinzipiell möglich, eine bestimmte Anzahl der Codierungen für 2-Byte-Sequenzen und den Rest für 3-Byte-Folgen und längere zu reservieren, jedoch ist das auch nicht gerade das Ei des Kolumbus. Die Codierung könnte dann beispielsweise folgendes Aussehen haben: Man codiert grundsätzlich zu 10 Bit. Dies ergibt bekanntlich 2 hoch 10 = 1024 mögliche Werte. 256 von diesen repräsentieren die 256 Byte, 512 stehen für die 512 häufigsten im zu packenden Programm vorkommenden 2-Byte-Sequenzen, 192 für 3-Byte- und 64 für 4-Byte-Folgen. Damit wird der Nachteil einer konstanten Sequenzlänge zwar aufgehoben, aber der dritte Nachteil noch verstärkt:

3) Gleich, welche Tricks man sich einfallen läßt, die ersten beiden Nachteile zu verringern, an teils ausgesprochen umfangreichen Tabellen, in denen die Sequenzen, die codiert werden sollen, abgelegt sind und die den gepackten

Code um ein beträchtliches Stück länger machen, führt kein Weg vorbei.

Uns wäre also sehr geholfen, wenn es ein Verfahren gäbe, das beim Entpackvorgang seine Tabellen selbst erzeugt.

Der gepackte Code würde also nicht durch hinzugefügte Tabellen um einiges verlängert. Des weiteren sollte die beim Codieren verwendete Codewortlänge flexibel sein (vergleiche Nachteil 1), und Sequenzen verschiedener Längen sollten ebenfalls codierbar sein (siehe Nachteil 2).

So erstaunlich das auch klingen mag – ein solches Verfahren gibt es wirklich. Es nennt sich, wie oben schon angedeutet, Lempel-Ziv-Verfahren.

Wir gehen dabei zur Vereinfachung zunächst von einer festen Codewortlänge von 12 Bit aus. Mit 12 Bit lassen sich 2 hoch 12 = 4096 Kombinationen darstellen, und zwar $000 (0) bis $FFF (4095).

Die Bytes $00 bis $FF werden im 12-Bit-Code einfach als $000 bis $0FF codiert. 4096 - 256 = 3840 12-Bit-Codes (von $100 bis $FFF) bleiben übrig. Diese verwenden wir für die Tabelle mit 3840 Einträgen T($100) bis T($FFF) zu je 20 Bit.

Ein Tabelleneintrag hat deshalb 20 Bit, weil wir, wie wir gleich an einem ausführlichen Beispiel sehen werden, in der Tabelle jeweils einen Tabellenindex (12 Bit) und ein Byte (8 Bit) eintragen können müssen.

Schritt	zu packender Code	Tabelle	Ausgabe (12 Bits)
1	aktuelles Byte (↓ $02) $00, $02, $33, $00, $02, $A2... $100 = $00, $02	Pos. V C $100 $000 $02	$000
2	aktuelles Byte (↓ $33) $00, $02, $33, $00, $02, $A5... $101 = $02, $33	Pos. V C $100 $000 $02 $101 $002 $33	$002
3	aktuelles Byte (↓ $00) $00, $02, $33, $00, $02, $A5... $102 = $33, $00	Pos. V C $100 $000 $02 $101 $002 $33 $102 $033 $00	$033
4	aktuelles Byte (↓ $A5) ...,$33, $00, $02, $A5, $00, $02... ≙$100 = $00, $02 $103 = $00, $02, $A5	Pos. V C $101 $002 $33 $102 $033 $00 $103 $100 $A5	$100
5	aktuelles Byte (↓ $00) ...,$00, $02, $A5, $00, $02, $00... $104 = $A5, $00	Pos. V C $102 $033 $00 $103 $100 $A5 $104 $0A5 $00	$0A5
6	aktuelles Byte (↓ $00) ...,$02, $A5, $00, $02, $00, $02... ≙$100 = $00, $02 $105 = $00, $02, $00	Pos. V C $103 $100 $A5 $104 $0A5 $00 $105 $100 $00	$100
7	aktuelles Byte (↓ $34) ...,$00, $02, $00, $02, $A5, $34 ≙$100 = $00, $02 ≙$103 = $00, $02, $A5 $106 = $00, $02, $A5, $34	Pos. V C $104 $0A5 $00 $105 $100 $00 $106 $103 $34	$103
8	letzter Code V (↓ $34) ...,$00, $02, $A5, $34	Keine weiteren Tabelleneinträge, da kein Code C mehr gelesen werden kann	$034

Bild 1. Die Funktionsweise des Lempel-Ziv-Packverfahrens

Die Tabelle sei zunächst »leer«. Sie wird im Laufe des Packvorganges »der Reihe nach gefüllt«, also zunächst wird in T($100) etwas eingetragen, dann in T($101) und so fort. Was das konkret heißt, wollen wir uns am Beispiel der Lempel-Ziv-Codierung der 12-Byte-Folge

$00, $02, $33, $00, $02, $A5, $00, $02, $00, $02, $A5, $34

nun einmal genauer ansehen.

Der Packer

Betrachten wir hierzu Bild 1. Im ersten Schritt werden zunächst dem zu komprimierenden Datensatz die ersten beiden Bytes entnommen; sie werden als Vorgängerbyte »V« und als Codebyte »C« in der Tabelle unter der nächsten freien Position ($100) eingetragen. Fortan ist dem Kompressor die 2-Byte-Folge $00, $02 kurz als 12-Bit-Code $100 bekannt. Ausgegeben – als gepackter Code also – wird grundsätzlich der zuletzt in die Tabelle eingetragene Vorgängercode V, hier also $000. Man bedenke, daß in unserem Fall ein Vorgängercode V immer 12 Bit lang ist. Den Grund dafür werden wir beim vierten Schritt erkennen. Doch zunächst weiter mit Schritt 2:

Alles, was im jeweils vorangegangenen Schritt als Vorgängercode diente und somit als gepackter Code ausgegeben wurde, interessiert uns nun nicht mehr. Das Codebyte C des letzten Schrittes ist ab jetzt der neue Vorgänger V und ein neues Code-Byte C muß den zu packenden Daten entnommen werden – hier also $33. Ab jetzt wird wie bei Schritt 1 verfahren; für den dritten Schritt gilt dasselbe wie für Schritt 2. Interessant wird es jedoch dann beim vierten Schritt. Der Vorgängercode V entspricht zunächst wieder dem Codebyte C des letzten Schrittes, ist hier also $000. Das neue Codebyte ist $02 – doch halt: ein Bytepaar mit V=$000 und C=$02 steht schon in unserer Tabelle unter

Position $100. Das bedeutet für den Packer letztendlich, daß das Paar $00, $02 ein einziger (Vorgänger-) Code, nämlich $100 ist, also V=$100, und C entspricht dem nächsten zu packenden Byte $A5. Da ein Code mit V=$100 und C=$A5 nirgends in der Tabelle zu finden ist, wird weiter wie bei den ersten drei Schritten verfahren. Für Schritt 5 gilt gleiches wie beispiesweise für Schritt 3, bei Schritt 6 ist analog zu Schritt 4 zu verfahren. Zum siebten Schritt ist jedoch noch einiges zu bemerken: Zunächst ist V hier wieder gleich $000 und C=$02. Dieser Code existiert bekanntlich in der Tabelle unter Position $100. Also ist $100 unser neuer Vorgängercode V und C=$A5. Die Werte $100 und $A5 wurden aber schon bei Schritt 4 in die Tabellenposition $103 eingetragen. Das bedeutet im Endeffekt, daß $103 für die 3-Byte-Sequenz $00,$02,$A5 steht.

Man sieht, mit dem Lempel-Ziv-Verfahren lassen sich auch 3-Byte-Sequenzen und längere zu 12 Bit codieren. In die Tabelle ist also $103, $34 einzutragen. Beim achten und letzten Schritt existiert nun nur noch ein Vorgängercode V=$034, jedoch kein Codebyte C, also wird der Vorgängercode wie gewohnt ausgegeben und der Codiervorgang ist beendet.

Der Entpacker

Sehen wir uns nun in Bild 2 an, wie ein zugehöriger Dekompressor arbeiten muß. Ihm liegen als gepackter Datensatz acht 12-Bit-Codes vor, und zwar:

$000, $002, $033, $100, $0A5, $100, $103, $034

In jedem Schritt unseres Entpack-Vorganges lesen wir einen gepackten Codewert ein, benötigen also insgesamt acht Schritte. Bei jedem Schritt wird der eingelesene 12-Bit-Code in die jeweils nächste freie Tabellenposition als Vorgängercode V eingetragen, decodiert und als ungepackter Code ausgegeben. Wie das Decodieren dieses Vorgängercode vor sich geht, soll hier nun an drei Beispielen näher erläutert werden:

Im Schritt 3 ist der eingetragene V-Code gleich $033, was bekanntlich für $33 steht. Zur Erinnerung: Codes von $000 bis $0FF stehen für die Bytes $00 bis $FF, alle weiteren Codes repräsentieren Tabellenpositionen und damit Byte-Paare, -tripel und so fort. Also ist $033 auch als $33 auszugeben. Interessanter ist da schon Schritt 4: der V-Code ist $100, entspricht also dem Inhalt der Tabellenposition $100 und steht somit für $000, $02. Da $000 natürlich $00 repräsentiert, stand $100 im Endeffekt für $00, $02. Dies wird auch als entpackter Code ausgegeben. Die Ausgabe in Schritt 7 ist noch tiefer verschachtelt. Der zu decodierende Code $103 steht – das entnehmen wir Tabellenposition $103 – für $100, $A5, dieser wiederum für $000, $02, $A5 und das bedeutet $00, $02, $A5. Man sieht deutlich, daß sich der Dekompressor bei der Bestimmung des entpackten Codes oftmals »rückwärts durch die Tabelle hangeln« muß. Bei der Realisierung dieser »Coderückverfolgung« verwendet man in der Regel einen Stack. Dies wollen wir uns am Beispiel des Decodierens von $103 im Schritt 7 einmal näher betrachten. Unter Tabellenposition $103 finden wir V=$100 und C=$A5. Wir legen C – also $A5 – auf dem (zunächst leeren) Stack ab und untersuchen Position $100. Wir erhalten V=$000 und C=$02. C – diesmal $02 – wird auf dem Stack abgelegt. V ist kleiner als $100 und somit kein weiterer Verweis auf einen Tabelleneintrag, sondern steht für $00, was auch seinem Platz auf dem Stack entspricht. Der Stackaufbau ist nun abgeschlossen; die Bytes wurden in der Reihenfolge $A5, $02, $00 abgelegt. Nun hat es ein Stack bekanntlich an sich, daß seine Elemente in umgekehrter Reihenfolge ausgelesen werden. Dies ergibt dann genau unsere entpackte Folge $00, $02, $A5.

Bei der Erklärung der Wirkungsweise des Entpackers muß natürlich noch erwähnt werden, wie die Tabelleneinträge C zustande kommen. Schauen wir uns hierzu der Illustration halber Schritt 6 an. Zunächst wird der V-Code $100 in Tabellenposition $105 eingetragen und dann (unter Zuhilfenahme eines Stacks) decodiert und ausgegeben. Das erste hierbei ausgegebene Byte ($00, das oben auf dem Stack liegt) wird nun als C unter der vorherigen (!) Tabellenposition $104 eingetragen.

Die Ausnahme von der Regel

Eins muß hier stutzig machen: Ein Decodieren und Ausgeben eines V-Codes einer Tabellenposition erfolgt noch vor dem Eintragen des C-Codes in die vorherige Tabellenposition. Wenn in der Tabelle an Position $104 ein Verweis (mittels V-Code) auf Position $103 erfolgt, dann ist der C-Code in Position $103 noch nicht eingetragen und der Stackaufbau ist zum Scheitern verurteilt. Doch kann dieser Fall überhaupt auftreten? Er kann, und zwar immer dann, wenn eine zu codierende Folge »BSBSB« auftritt, wobei »B« ein Byte und »S« eine Sequenz von mindestens einem Byte Länge ist. Allerdings muß die Sequenz »BS« weiter vorne in der Tabelle schon einmal als Code festgehalten worden sein. Dies macht man sich am besten anhand eines Beispieles klar. Wir wählen B=$11 und eine 1-Byte-Sequenz S=$22. Außerdem müssen wir garantieren, daß BS=$11, $22 vor dem »BSBSB-Fall« schon als Code in der Tabelle eingetragen ist – am besten, wir lassen unseren zu packenden Datensatz damit beginnen. Dieser könnte dann zum Beispiel folgendes Aussehen haben:

$11, $22, $00, $11, $22, $11, $22, $11.

Zum Zwecke der Übung des Lempel-Ziv-Packens sollten Sie diese 8-Byte-Folge einmal codieren. Ihr Ergebnis sollte aus fünf 12-Bit-Werten bestehen und wie folgt lauten: $011, $022, $000, $100, $103.

Übung macht den Meister

Der Darstellung in Bild 3 können Sie dem Entpack-Vorgang entnehmen. Das Dekomprimieren erfolgt hier analog zu unserem ersten Entpack-Beispiel in Bild 2. Die »BSBSB«-Ausnahme ist in Schritt 5 klar ersichtlich: Unter Tabellenposition $104 wird zunächst der V-Wert $103 eingetragen. Dieser stellt einen Verweis auf die voherige, zu diesem Zeitpunkt noch unvollständige Position $103 dar. Wir wissen jedoch, daß es sich in einem solchen Falle nur um eine »BSBSB«-Folge handeln kann; bei dem fehlenden C-Wert in Position $103 muß es sich also um B und somit um $11 handeln. Die Behandlung dieses Ausnahmefalles in einen Lempel-Ziv-Dekompressor einzubinden, sieht auf den ersten Blick sicher komplizierter aus als sie ist. Es empfiehlt sich, noch vor dem Eintragen des V-Codes in eine Tabellenposition (zum Beispiel $104) den C-Code der vorherigen Position ($103) einzutragen, indem man einfach den C-Code der vorletzten Position ($102) übernimmt. Zur Überprüfung des eben Gesagten betrachten wir zwei Fälle:

1) Es tritt der »BSBSB«-Fall auf. In diesem Falle wird zu Beginn von Schritt 5 der richtige Wert $11 in den C-Code der Position $103 übernommen, dann der V-Wert $103 richtig decodiert und zuletzt wird $11 – da als erstes decodiertes Byte in Schritt 5 ausgegeben – in den C-Wert von $103 übernommen, wo vorher ja auch schon $11 stand.

2) Der Sonderfall tritt nicht auf, beispielsweise in Schritt 4. Hier würde in den C-Wert von Position $102 zunächst der C-Wert von $101 übertragen – also $00. Da der »BSBSB«-

Fall nicht auftritt, gibt es auch keinen Verweis (mittels V-Code) von Position $103 nach $102. Demnach ist der C-Wert in Position $102 zu diesem Zeitpunkt ohnehin beliebig.

Natürlich funktioniert ein Übernehmen des C-Wertes aus der vorletzten in die letzte Position erst ab Tabellenposition $102, vorher kann der Sonderfall jedoch ohnehin nicht auftreten.

Variable Codewortlängen helfen Platz sparen

Nachdem nun die Wirkungsweise eines LZ-Packers und -Entpackers klar geworden sein dürfte, wollen wir uns nun einmal ansehen, welche der oben beschriebenen Nachteile er nicht in sich birgt. Tabellen müssen offenbar nicht mit dem gepackten Code (und dem Dekompressor) gespeichert werden, da sie beim Entpacken selbst erzeugt werden. Auch werden Sequenzen, bestehend aus 2, 3, 4 oder noch mehr Byte, codiert. Nur die Codewortlänge ist nicht flexibel (vergleiche Nachteil 1), sondern konstant 12. Doch das läßt sich leicht ändern.

Mit 12-Bit-V-Werten kann man Verweise auf Tabellenpositionen bis $FFF (4095) erreichen. Die Tabellen in unseren Beispielen waren jedoch weitaus kleiner. Für Tabellenpositionen bis $1FF reichen natürlich auch 9-Bit-V-Codes, bis $3FF genügen 10-Bit-Werte, bis $7FF 11-Bit Werte und so weiter. Das bedeutet: Solange beim Kompressen die Tabelle noch nicht bis zu Position $200 (=2 hoch 9) angewachsen ist, wird der gepackte Code jeweils als 9-Bit-Wert ausgegeben, solange die Tabelle nicht auf $400 (= 2 hoch 10) angewachsen ist, werden 10-Bit-Codes ausgegeben und so fort. Da der Dekompressor analog zum Kompressor die Tabelle aufbaut, gilt für ihn das gleiche für einzulesende Codes, das heißt er liest erst eine Zeitlang 9-Bit-Werte, dann 10-Bit-Werte .....

In unserem ersten Beispiel (Bild 1) hätte also eine 9-Bit-Ausgabe völlig ausgereicht, es wären also 8 Codes zu je 9 Bit = 72 Bit = 9 Byte ausgegeben worden, was bei einem Eingabecode von 12 Byte 25 Prozent Einsparung bringt.

Der Code zum Tabellenlöschen

Beim Packen wird immer dann, wenn eine schon in der Tabelle verzeichnete Sequenz gefunden wird, etwas eingespart, ansonsten etwas eingebüßt. Da zu Beginn aufgrund der noch ziemlich leeren Tabelle in der Regel wenig gefunden wird, verliert man zunächst einmal einiges, holt den Verlust aber mit zunehmender Tabellengröße schnell wieder ein und kommt somit also in einen Bereich, wo sich ein LZ-Packen lohnt.

Mit weiter zunehmender Tabellengröße werden zwar auch immmer häufiger in der Tabelle eingetragene Sequenzen gefunden, jedoch steigt auch die Codewortlänge an. Bei einer Codewortlänge von 12 Bit liegt die aktuelle Tabellenposition wie im letzten Absatz angedeutet zwischen $800 und $FFF, die Tabelle hat also zwischen 1792 und 3839 Einträge. Somit hat man zwar gute Chancen, eine Sequenz in der Tabelle zu finden, wenn man Pech hat, muß man jedoch einen 8-Bit-Eingabecode zu einem immerhin 12 Bit langen Ausgabecode codieren und verliert 4 Bit. Bis zu einer bestimmten, vom zu packenden Programm abhängigen Tabellengröße lohnt sich ein Lempel-Ziv-Packen besonders, nach diesem Punkt wird die »Packausbeute« geringer. Deshalb kann es – besonders bei sehr langen zu packenden Programmen – sehr nützlich sein, beim Kompressionsvorgang ein Protokoll über die jeweilige »Ausbeute« zu führen. Dies kann beispielsweise geschehen, indem man in regelmäßigen Abständen das Verhältnis von (ungepacktem) Eingabecode und (gepacktem) Ausgabecode ermittelt und, wenn das Ergebnis schlechter wurde als beim letzten Test, vom nächsten Eingabecode an quasi von vorne mit dem Packvorgang beginnt, also die Tabelle löscht und wieder mit einer Codewortlänge von 9 beginnt. Der Kompressor muß dem Dekompressor natürlich mitteilen, zu welchen Zeitpunkt auch jener seine Tabelle zu löschen hat.

Schritt	gepackter Code	Tabelle Pos.	Tabelle V	Tabelle C	Ausgabe (entpackter Code)
1	$000	$100	$000		$00
2	$002	$100 $101	$000 $002	$02 	$02
3	$033	$100 $101 $102	$000 $002 $033	$02 $33 	$33
4	$100	$101 $102 $103	$002 $033 $100	$33 $00 	$00 $02
5	$0A5	$102 $103 $104	$033 $100 $0A5	$00 $A5 	$0A5
6	$100	$103 $104 $105	$100 $0A5 $100	$A5 $00 	$00 $02
7	$103	$104 $105 $106	$0A5 $100 $103	$00 $00 	$00 $02 $A5
8	$034	$105 $106 $107	$100 $103 $034	$00 $34 	$34

Bild 2. Die Wirkungsweise des Dekompressors

Schnellerer Tabellenzugriff beim Packen

Dies geschieht am sinnvollsten dadurch, daß man zwischen Kompressor und Dekompressor einen Tabellenlösch-Code vereinbart, zum Beispiel $100. Immer, wenn der Entpacker ein $100 als Eingabecode liest, löscht er die Tabelle und beginnt auch mit seinem Tabellenaufbau von neuem.

Selbstverständlich beginnt die Tabelle damit nicht mehr bei Position $100, sondern bei $101.



Beim oben beschriebenen Lempel-Ziv-Packer fällt auf, daß, vor allem bei größeren Tabellen, sehr viel Zeit für das Suchen von Sequenzen in der Tabelle aufgewendet werden muß. Es sei deshalb hier angemerkt, daß der Tabellenaufbau des Kompressors nicht unbedingt sequentiell erfolgen muß, sondern, daß auch ein sogenanntes Hashing-Verfahren gewählt werden kann, dessen Zugriffsschlüssel abhängig von den einzutragenden V- und C-Werten bestimmt wird. Je größer dabei der freie RAM-Speicherplatz für die Tabelle ist, desto schneller arbeitet der Packer.

Schritt	gepackter Code	Tabelle	Ausgabe
1	$011	Pos. / V / C $100 / $011 /	$11 (B)
2	$022	Pos. / V / C $100 / $011 / $022 $101 / $022 /	$22 (S)
3	$000	Pos. / V / C $101 / $022 / $00 $102 / $000 /	$00
4	$100	Pos. / V / C $102 / $000 / $11 $103 / $100 /	$11 (B) $22 (S)
5	$103	Pos. / V / C $102 / $000 / $11 $103 / $100 / $?? $104 / $103 /	$11 (B) $22 (S) ? (B)

Bild 3. Die Ausnahme beim LZ-Dekompressionsvorgang

Andere Pack-Verfahren

Zum Abschluß dieses Grundlagenartikels wollen wir nun noch einen kurzen Ausblick auf andere, hier nicht näher beschriebene Verfahren wagen. Alle bis hierher vorgestellten Verfahren haben eines gemeinsam: Es existiert ein Umkehralgorithmus, und damit ist auch ein Dekompressor möglich, der in der Lage ist, den gepackten Code wieder in seinen ursprünglichen Zustand zurückzuversetzen.

Man kann sich aber auch Fälle denken, in denen es nicht notwendig ist, daß der Entpacker exakt den ungepackten Ausgangscode wiederherstellt. Dies ist zum Beispiel bei einem Datensatz aus digitalisierten Sounddaten der Fall. Unter Umständen kann es genügen, daß eine digitalisierte Musik nur »sehr ähnlich« klingt, nachdem man sie ge- und entpackt hat. Auf derartige Verfahren wollen wir — schon aus Platzgründen — nicht näher eingehen, dies sollte uns nur als Beispiel dafür dienen, daß es auch anders geht.

(Peter Arndt/Ulrich Völker/sk)

Literaturhinweise:

J.Ziv, A.Lempel: A Universal Algorithm for Sequential Data Compression, IEEE Trans. Information Theory, Vol.IT-23, No.3, May 1977,pp 337-343.

Fibonacci Delta Compression, AMIGA ROM Kernel Reference Manual: Exec, Addison-Wesley, p. B-68.

Genau betrachtet: RS232/V.24-Schnittstelle

Eine kurze und bündige Beschreibung der RS232-Schnittstelle Ihres C 64. Was machen die Signale, wie sind die Pin-Belegungen?

Bei der RS232-Schnittstelle werden die Daten Bit für Bit übertragen, im Gegensatz zur Centronics- oder IEEE-488-Norm, bei der ganze Bytes übergeben werden. Die Bits werden als eine Folge von Spannungsimpulsen mit einer bestimmten Dauer übertragen. In der Praxis werden dabei Pakete von 5 bis 8 Datenbit übertragen, die von einem Start-Bit und 1 bis 2 Stop-Bit eingerahmt sind (Bild 1). Das Start-Bit hat grundsätzlich logischen Low- und die Stop-Bits High-Pegel. Vor dem Stop-Bit kann ein sogenanntes Paritäts-Bit vereinbart werden, das die Anzahl der High-Zustände im Datenwort immer gerade oder ungerade macht.

Beispiel: Sind in einer 8-Bit-Übertragung 5 Bit gesetzt, wird das Paritäts-Bit ebenfalls gesetzt, wenn gerade Parität vereinbart wurde.

Um die Störungs-Anfälligkeit der Übertragung zu mindern, wird logisch »Eins« (gesetztes Bit) nicht durch +5V (TTL-Pegel) realisiert, sondern mit einer Spannung von —3 bis —12V und logisch »Null« mit +3 bis +12V (RS232 nach DIN 66020). Der C 64 hat zwar die nötige Software für eine RS232-Schnittstelle im Betriebssystem integriert, verfügt aber nicht über die entsprechenden Spannungspegel. Im C 64 gibt es nur zwei Spannungen: +5V (TTL) und 9V Wechselspannung. Es ist also ein Interface zur Spannungskonvertierung nötig. Links in Bild 2 finden Sie zwei Schaltpläne für ein solches Interface. Rechts im Bild die diskrete Lösung, für die Konvertierung von 0/5V auf ±12V (oben) und von ±12V auf 0/5V (unten). Beachten Sie, daß

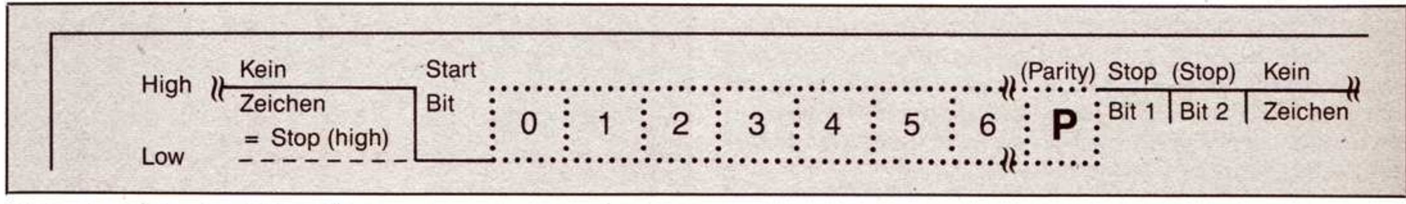


Bild 1. So sieht eine RS232-Übertragung schematisch aus. Das Start-Bit ist immer »Low«.

jede Sende- und Empfangsleitung die entsprechende Transistorschaltung braucht.

Mit einer Masse- und einer Datenleitung könnte schon eine Übertragung von Texten an einen Drucker erfolgen. Was ist aber, wenn die Datenübertragung schneller ist, als der Drucker die Zeichen zu Papier bringen kann? Der Drucker muß dem Computer sagen, wenn er zuviel Arbeit bekommt. Er kann dies auf drei verschiedene Arten tun:

Daten aufnehmen, sendet er ein ACK an den Computer und die Übertragung beginnt von Neuem.

— Hardware-Protokoll

Spätestens hier wird es unübersichtlich. Es hilft nur noch Tabelle 1 zur Erklärung der ganzen Signale. Als üblicher Stecker für V.24-Signale hat sich ein 25poliger D-Sub-Stecker (im Laborslang Cannon genannt) durchgesetzt. Die Bezeichnung der Kontakte ist gleich dreimal genormt:

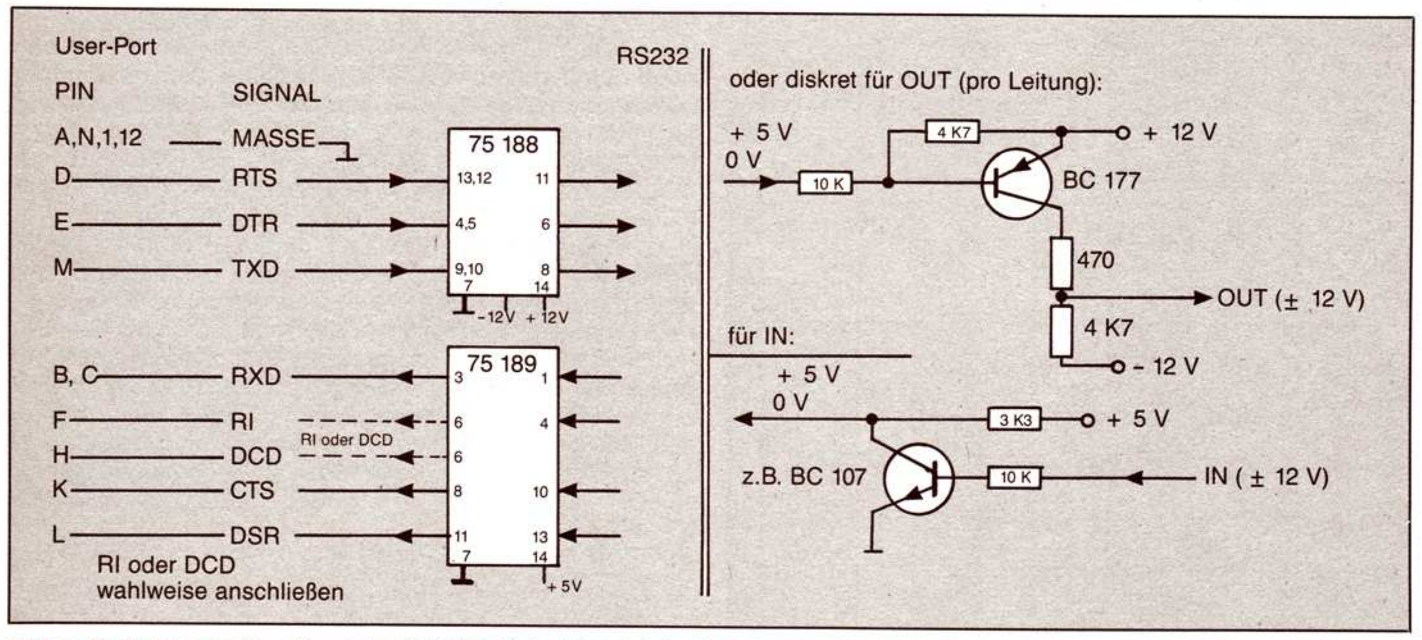


Bild 2. Zwei Schaltpläne für ein RS232/V.24-Interface. Links die Lösung mit IC, recht die diskrete mit Einzelelementen

— Software-Protokoll mit XON/XOFF

Es wird eine zusätzliche Leitung zwischen Computer und Peripherie eingerichtet, über die das empfangende Gerät den Code $13 (XOFF) sendet, wenn es keine Daten mehr annehmen kann. Dieses Signal hat die gleiche Aufgabe wie die Busy-Leitung einer Centronics-Schnittstelle; es stoppt die Datenübertragung. Die Freigabe erfolgt mit dem Code $11 (XON). Die Codes $11 und $13 entsprechen den ASCII-Codes DC1 und DC2.

Die neue Leitung kann natürlich auch zur Übertragung von mehr Informationen verwendet werden. Sende- und Empfangsgerät müssen dann allerdings in der Lage sein, zwei Leitungen (XON/XOFF und die normale Datenleitung) gleichzeitig zu verwalten. Mit dieser zusätzlichen Leitung wird auch der sogenannte Vollduplex-Betrieb möglich. Vollduplex heißt, daß beide Geräte gleichzeitig senden oder empfangen können. Im Gegensatz zum Halbduplex-Betrieb, bei dem zur gleichen Zeit nur in eine Richtung übertragen werden kann.

— Software-Protokoll mit ETX/ACK

Auch bei dieser Lösung kommt man nicht ohne eine zusätzliche Leitung aus. Sie heißt DTR (Data Terminal Ready). Ist zum Beispiel der angeschlossene Drucker bereit, Daten anzunehmen, aktiviert er die DTR-Leitung und sendet $06 (ACKnowledge). Der Computer schickt nun eine Reihe Datenworte über die Sendeleitung und schließt die Übertragung mit $03 zwischendurch immer wieder ab. Den nächsten Datenblock sendet er erst dann, wenn der Drucker sein ACK gegeben hat. Damit die Übertragung nicht in einem Chaos entartet, muß der Sende-Computer über das Puffervermögen des Empfängers informiert sein, um rechtzeitig ein ETX (End Of Text) senden zu können. Nur so kann ein Überlauf des Puffers und der damit einhergehende Datenverlust verhindert werden.

Hat das Empfangsgerät ein ETX festgestellt, werden die empfangenen Daten bearbeitet. Kann der Empfänger neue

DIN 66020, CCITT V.24 (Comité Consultatif International Télégraphique et Téléphonique) und EIA RS232C (Electronic Industries Association). Die Bedeutung der Signale ist bei allen Normen gleich, nur die Signalpegel differieren. Die deutsche Norm verlangt, im Gegensatz zu den anderen, negative Logik.

Zum Anschluß einer RS232/V.24-Schnittstelle ist es in den meisten Fällen nicht nötig, alle Leitungen zu benutzen. So werden nur wenige unter Ihnen eine synchrone Daten-

Pin/Bedeutung	DIN	CCITT	EIA	User-Port C 64 (VC 20)
1 Masse	E1	101	AA	A-GND (A-GND)
2 Transmit data (TD) Über diese Leitung sendet der C 64 Daten an den Akustikkoppler.	D1	103	BA	M-PA2 (M-CB2) out
3 Received data (RD) Die Empfangsleitung.	D2	104	BB	B-F12 + C-PB0 (B-CB1 + C-PB0)
4 Request to send (RTS) Frage des Computers an das Peripheriegerät, ob es zur Datenübertragung bereit ist.	S2	105	CA	D-PB1 (D-PB1) out
5 Clear to send (CTS) Positive Antwort des Peripheriegerätes auf RTS.	M2	106	CB	K-PB6 (K-PB6)
6 Data set ready (DSR) Akustikkoppler ist betriebsbereit	M1	107	CC	L-PB7 (L-PB7)
7 Signalmasse	E2	102	AB	N-GND (N-GND)
8 Received line signal (DCD)	M5	109	CF	H-PB4 (H-PB4)
9 Testzwecke				
10 Testzwecke				
11 nicht belegt				
12 Secondary carrier detector	HM5	122	SCF	—
13 Secondary clear to send	HM2	121	SCB	—
14 Secondary transmitted Data	HD1	118	SBA	—
15 Transmit clock (TC) from modem	T2	114	DB	—
16 Secondary received data	HD2	119	SBB	—
17 Receiver signal clock (RC)	T4	115	DD	—
18 nicht belegt				
19 Secondary request to send	HS2	120	SCA	—
20 Data terminal ready (DTR) Terminal ist zur Datenübertragung bereit.	S1.x	108.x	CD	E-PB2 (E-PB2) out
21 Signal quality detector	M6	110	CG	—
22 Ring indicator (RI)	M3	125	CE	F-PB3 (F-PB3)
23 Data signal rate det. terminal	S4	111	CH	—
modem	M4	112	CI	—
24 Transmit clock to modem	T1	113	DA	—
25 nicht belegt				

Tabelle 1. Belegung der RS232-Leitungen.

übertragung mit zusätzlichem Clock-Signal realisieren, wie beim seriellen IEC-Bus des C 64. Normalerweise reichen die folgenden Leitungen aus:

1. eine Masseleitung
2. je Richtung eine Datenleitung
3. je Richtung eine Busyleitung

Die Punkte 1 und 2 dürften klar sein. Punkt 3 kann auf vielfältige Art realisiert werden. In aller Regel werden die Kontakte S2 (RTS) und M2 (CTS) benutzt (siehe auch Tabelle 1). Die Erklärung erfolgt am besten an einem Beispiel: Der Drucker zeigt seine Empfangsbereitschaft an, indem er M2 aktiviert. Dieses Signal fragt der Computer ständig am Anschluß S2 ab. Ist M2 inaktiv, stoppt der Computer die Datenübertragung. Nehmen wir an, anstelle des Druckers sei ein Meßgerät angeschlossen, das nur ab und zu Anweisungen vom Computer bekommt und ansonsten sich um interne Aufgaben kümmert. Dann wäre es wenig vorteilhaft, wenn das Meßgerät ständig Befehle vom Computer erwartet; für die eigentlichen Meßaufgaben bliebe zu wenig Zeit. In diesem Fall gestattet das einfache Abfragen des Kontaktes S1 (per Interrupttechnik) eine fast ungestörte Bearbeitung eines Programmes. S1 ist mit RTS (Request To Send) identisch. Aktiviert der Computer S1, »spitzt das Meßgerät die Ohren« und das Meßprogramm verzweigt in die Datenempfangsroutine.

Nicht auf die Norm verlassen

Die Hersteller von V.24-Schnittstellen scheinen sich nicht immer völlig einig zu sein, wie die Belegung und Bedeutung der einzelnen Stecker-Pins ist. So sind diesem Beitrag hauptsächlich Praxiserfahrungen zugrunde gelegt. Besser als Normblätter ist die Überprüfung der Schnittstelle mit einem Speicheroszilloskop oder einem Digital-Analyzer. Zumal die Steuerleitungen ab und zu mit Fantasienamen belegt oder nicht eindeutig als Sende- oder Empfangsleitung gekennzeichnet werden. Zum störungsfreien Betrieb einer Schnittstelle sollten nichtbenutzte Leitungen auf ein festes Potential gelegt werden. Man verhindert dadurch, daß die Übertragung bei einer eventuellen Abfrage einer solchen Leitung, mit undefiniertem logischen Zustand, nicht unterbrochen wird.

Beim C 64 kann diese Schnittstelle mit Gerätenummer 2 angesteuert werden. Zur Bestimmung der Kontrollparameter sind zwei Register vorhanden, die auch von Basic aus erreicht werden können.

Einstellparameter

Mit Tabelle 2 können Sie die Werte bestimmen, die Sie in Kommando- und Kontrollregister schreiben müssen, um ein bestimmtes Übertragungsprotokoll zu bewerkstelligen. Eine »1« bedeutet ein gesetztes Bit.

Das Einschalten der RS232-Schnittstelle erfolgt beim C 64 mit OPEN filenr.,2,0,CHR$(Kontrollreg.)+CHR$(Kommandoreg.).

Beispiel: OPEN20,2,0,CHR$(64+4+2)+CHR$(32+1).

Mit dieser Anweisung wird für Filenummer 20 vereinbart: 1 Stop-Bit, 6 Datenbit, 300 bit/s, ungerade Parität, Vollduplex und Hardwareprotokoll.

Wird eine Filenummer größer als 128 verwendet, sendet der Computer nach jedem Carriage Return ($0D) ein Linefeed ($0A). Sollten diese Einstellmöglichkeiten nicht ausreichen, können Sie direkt in die RS232-Routine eingreifen. In der Tabelle 3 finden Sie eine Übersicht der Speicherstellen, die von der RS232 benutzt werden.

Auf der Seite 71 finden Sie eine softwaremäßige RS232-Steuerung, mit der bis zu 4800 bit/s übertragen werden können.

Fehlerabfrage

Das Betriebssystem des C 64 verfügt über eine Fehlerbehandlung der RS232-Schnittstelle. Der Status dazu kann entweder durch die Variable ST abgefragt werden oder direkt mit Speicherzelle $0297. Die Bedeutung der einzelnen Bits von ST finden Sie in Tabelle 4. (Jens Maßmann/kn)

1.) Kontrollregister bzw. Steuerregister ($293)

Bit (dez. Wert)	Bedeutung
Bit 7 (128)	0 = 1 Stop-Bit, 1 = 2 Stop-Bits
Bit 4 (16)	nicht benutzt

Bit (dez. Wert)	8 Daten-Bits	7 Daten-Bits	6 Daten-Bits	5 Daten-Bits
Bit 6 (64)	0	0	1	1
Bit 5 (32)	0	1	0	1

Übertragungsraten (Bit/sec)

Bit (dez. Wert)	nicht impl.	50	75	110	134,5	150	300	600	1200	1800	2400
Bit 3 (8)	0	0	0	0	0	0	0	0	1	1	1
Bit 2 (4)	0	0	0	0	1	1	1	1	0	0	0
Bit 1 (2)	0	0	1	1	0	0	1	1	0	0	1
Bit 0 (1)	0	1	0	1	0	1	0	1	0	1	0

2.) Kommandoregister bzw. Befehlsregister ($ 294)

Bit (dez. Wert)	keine Paritätsüberprüfung (alle vier Komb.)				ungerade Parität	gerade Parität	Bit 8:1 ohne Parität	Bit 8:0 ohne Parität
Bit 7 (128)	0	0	1	1	0	0	1	1
Bit 6 (64)	0	1	0	1	0	1	0	1
Bit 5 (32)	0	0	0	0	1	1	1	1

Bit (dez. Wert)	Bedeutung
Bit 4 (16)	0 Vollduplex, 1 Halbduplex
Bit 3 (8)	nicht benutzt
Bit 2 (4)	nicht benutzt
Bit 1 (2)	nicht benutzt
Bit 0 (1)	0 Freilaufmodus (3-Draht), 1 Hardwarehandshake (X-Draht)

◄ **Tabelle 2. Die Funktion des Kontroll- und Kommandoregisters auf einen Blick**

Bit	dez.	Bedeutung
0	1	Paritätsfehler
1	2	Rahmenfehler
2	4	Empfängerpuffer voll
3	8	unbenutzt
4	16	CTS-Signal fehlt
5	32	unbenutzt
6	64	DSR-Signal fehlt
7	128	Break-Signal empfangen

Tabelle 4. Bedeutung der Statusvariablen ST beziehungsweise der Speicherstelle $0297

RS232-Schnittstelle		
167	$A7	Zwischenspeicher für Eingabe über die RS232-Schnittstelle
168	$A8	Bitzähler für RS232-Eingabe
169	$A9	RS232-Flagge für Start-Bit-Prüfung
170	$AA	RS232-Eingabespeicher
171	$AB	Parityprüfung
181	$B5	RS232-Anzeige für nächstes Bit
182	$B6	Ausgabe-Zwischenspeicher für RS232
189	$BD	Zwischenspeicher für RS232-Parity-Prüfung
247-248	$F7-$F8	Zeiger auf den Anfang des RS232-Eingabepuffers
249-250	$F9-$FA	Zeiger auf den Anfang des RS232-Ausgabepuffers
659	$293	RS232-Steuerregister (Kontrollregister)
660	$294	RS232-Befehlsregister (Kommandoregister)
661-662	$295-$296	RS232 frei wählbare Übertragungsrate
663	$297	RS232-Statusregister
664	$298	RS232-Anzahl der zu übertragenden Bits
665-666	$299-$29A	Zeit, die bei RS232 zum Übertragen eines Bits gebraucht wird (Übertragungsrate)
667	$29B	Index auf das Ende des RS232-Eingabepuffers
668	$29C	Index auf den Anfang des RS232-Eingabepuffers
669	$29D	Index auf den Anfang des RS232-Ausgabepuffers
670	$29E	Index auf das Ende des RS232-Ausgabepuffers

Tabelle 3. Die Adressen, welche für die Datenübertragung von der RS232-Schnittstelle benötigt werden



Geschwindigkeit in Basic

Ist 740914799 eine Primzahl? Um dieses Problem zu lösen, ehe Sie in Rente gegangen sind, benötigen Sie ein ausgeklügeltes Programm. Wie ein solches Programm schrittweise verfeinert und beschleunigt wird, wollen wir Ihnen zeigen, denn auf das Prinzip kommt es an.

Natürlich kann man so etwas auch in Maschinensprache lösen, sogar 100- bis 1000mal schneller als in Basic. Allerdings kann die Umsetzung vieler Probleme in Maschinensprache auch 100mal so aufwendig sein, gerade wenn mit Variablen oder reellen Zahlen gearbeitet werden muß. Einfacher und oft auch ausreichend schnell ist die Optimierung des Basic-Programms.

Gleich der erste Tip:

Nutzen Sie die maximale Länge einer Basic-Zeile und versuchen Sie, FOR..NEXT-Schleifen in einer Basic-Zeile unterzubringen.

Als Beweis dient das folgende kleine Basic-Programm.

```
100 FOR X = 1 to 10000 : Y = X + X : NEXT
```

Dieses Programm benötigt 22 Sekunden auf dem C 128 beziehungsweise 34 auf dem C 64 (die Art der Zeitmessung wird später noch besprochen). Teilen Sie dieses Programm in drei Zeilen auf,

```
100 FOR X = 1 to 10000
110 Y = X + X
120 NEXT
```

so steigt die Zeit auf 23 beziehungsweise 35 Sekunden. Zur Erklärung: Der Basic-Interpreter kopiert beim Abarbeiten eines Basic-Programms eine Basic-Zeile in einen Puffer, interpretiert den Inhalt und führt die Befehle aus. Danach wird die nächste Zeile kopiert und so weiter. Sie können sich denken, warum das erste Beispiel schneller ist. Der Interpreter mußte die Zeile nur ein einziges Mal kopieren. Beim zweiten Beispiel wird erst Zeile 100 kopiert und ausgeführt, dann Zeile 110 und zum Schluß die Zeile 120. Der Befehl NEXT in Zeile 120 bewirkt einen Rücksprung zur Zeile 100, die dann erst wieder kopiert wird. Das heißt also, jede einzelne Zeile des zweiten Beispiels muß 10000mal in den Puffer kopiert werden, während der Einzeiler als Beispiel 1 nur ein einziges Mal zu kopieren ist.

Sicher sagen Sie, daß man noch mehr Zeit einsparen kann, wenn die Leerzeichen weggelassen werden, denn auch das Kopieren von Leerzeichen kostet Zeit. Das ist auch korrekt, aber der Erfolg läßt sich allerhöchstens bei Beispiel 2 feststellen, da diese Zeilen ja mehrmals kopiert werden. Die Leerzeichen sollten hier auch nur zur besseren Lesbarkeit beitragen.

Wir wollen uns hier aber nicht mit solchen Spitzfindigkeiten beschäftigen, sondern es sollen verschiedene Lösungswege zu bestimmten Problemen entwickelt und optimiert werden.

Natürlich wird man damit nicht annähernd an die Geschwindigkeit der Maschinensprache heranreichen können. Aber die Zeiteinsparungen liegen weit über denen des obigen Beispiels.

Die Problemstellung könnten Sie aufgrund des Vorspanns schon erahnen. Es soll hier um die Berechnung von Primzahlen gehen. Für diejenigen, deren Schulzeit schon etwas weiter zurück liegt, hier eine Erläuterung aus dem Lexikon.

Primzahl, in der Mathematik eine natürliche Zahl, die keine andere natürliche Zahl als Teiler hat, außer 1 und sich selbst. Die ersten Primzahlen sind 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, ... Jede natürliche Zahl ist entweder prim oder kann als Produkt von Primzahlen dargestellt werden, zum Beispiel 780 = 2 * 2 * 3 * 5 * 13. Es gibt unendlich viele Primzahlen, obwohl sie mit wachsendem Betrag immer spärlicher werden.

Da Sie jetzt wissen, worum es gehen soll, hier die erste Problemstellung:

Schreiben Sie ein Programm, das alle Primzahlen, beginnend mit 2, bis zu einer eingegebenen Grenze berechnet.

Die Realisierung dieses Problems dürfte eigentlich nicht schwer sein. Bevor wir uns aber der Lösung widmen, hier noch einige Hinweise zur Zeitmessung: Alle Listings sind

auf einem C 128, aber in Basic V2.0 erstellt worden. Das heißt, daß alle Programme auch auf dem C 64 lauffähig sind. Die Zeitmessung erfolgt mit der internen Uhr TI$. Für den Vergleich unserer Lösungen untereinander reicht dies völlig aus. Die Uhr wird mit TI$= "000000" gleich nach der Eingabe der Obergrenze gelöscht. Die Ausgabe der Zeit erfolgt nach dem letzten Basic-Befehl mit PRINT TI$. Die angegebene Zeit bezieht sich immer auf den C 128 im FAST-Modus, während die Zeit in Klammern für den C 64 gilt.

So, nun kommen wir zu unserem Programm zurück. Vielleicht haben Sie auch schon eine Lösung gefunden, die dem Programm »Prim-Rechner 1a« (Listing 1) ziemlich nahe kommt.

Hier nun die Programmbeschreibung:

100: Einlesen der Obergrenze und Ausgabe der 2 als erste Primzahl.

110: Zwei Zählschleifen mit X und Y. Die X-Schleife zählt die Zahlen, die getestet werden sollen, während die Y-Schleife den Dividend, der von 2 bis X-1 läuft, enthält. Somit wird X durch alle möglichen Teiler dividiert.

120: Treten bei der Division keine Nachkommastellen auf, ist die betreffende Zahl X durch Y teilbar und somit keine Primzahl. Deswegen wird nach 140 verzweigt.

130: Die Y-Schleife wird erhöht. Ist sie ganz durchgelaufen, das heißt X war nie durch Y teilbar, so handelt es sich um eine Primzahl und die Zahl wird ausgegeben.

140: Erhöhung der X-Schleife für die nächste zu prüfende Zahl.

Hinweis: Die Zahl 2 wird deshalb vorher ausgegeben, weil die zweite Schleife erst mit dem Wert 2 beginnt. Folglich wäre die 2 nicht als Primzahl erkannt worden.

Das Programm benötigt die folgenden Bearbeitungszeiten (siehe auch Tabelle 1).

Primzahlen bis:	50	100	200
Zeit in Sek.:	2	8	33
	(5)	(16)	(63)

Wollten Sie dieses Programm bis 740914799 rechnen lassen, so würden Sie das Ergebnis auf dieser Welt nicht mehr erfahren; bis dahin benötigt der C 128 $3,7 * 10^{44}$ Sekunden oder $1,2 * 10^{37}$ Jahre. Zum Vergleich: die Erde existiert erst seit zirka $5 * 10^{9}$ Jahren. Sie sehen, das Programm ist zu langsam, und es soll nun unsere Aufgabe sein, das Programm Prim-Rechner 1a so zu verändern, daß sich die Berechnungszeiten drastisch verringern.

Wir müssen herausfinden, wo Zeit zu sparen ist. Dazu spielen wir das Programm einmal mit zwei verschiedenen X-Werten durch.

Zeitentabelle 1

angegeben in Sekunden für C128 bzw. C64

PRG. NAME	BIS 50	BIS 100	BIS 200	BIS 500	BIS 1000	BIS 2000	BIS 5000	BIS 10000	BIS 4E05	GRENZE
PR1A	2 5	8 16	33 63							
PR1B	1 3	5 9	18 34	87 165						
PR1C	1 2	2 5	7 13	21 40	49 95					
PR1D	1 2	2 4	5 9	14 28	33 63	76 147				
PR1E	1 2	2 4	4 9	14 26	31 60	72 137				

PR = Prim - Rechner

Gesetzt den Fall, X beträgt 21 beim Erreichen der Y-Schleife. Durch 2 ist 21 nicht teilbar, deshalb wird Y auf 3 erhöht. Da 21 durch 3 teilbar ist, wird in Zeile 120 aus der Schleife herausgesprungen und die nächste Zahl (in diesem Fall 22) untersucht. Für die Ermittlung von 21 als Nicht-Primzahl wurden nur zwei Rechnungen benötigt. Spielen wir dies Programm einmal mit dem Wert 23 durch. Die Y-Schleife läuft von 2 bis X-1 (=22). Wenn man die Rechnung X/Y mit dem Taschenrechner nachvollzieht, stellt man fest, daß ab Y=12 das Ergebnis kleiner als 2 ist. Das heißt, daß eine Zahl nur durch Zahlen, die kleiner als die Hälfte der Zahl sind, restlos teilbar ist. Dies ist natürlich keine neue mathematische Erkenntnis, aber für uns ist dies sehr interessant, da wir nun unsere Y-Schleife nur bis zur Hälfte von X laufen lassen müssen.

```
10 REM ********************************** <060>
20 REM *        PRIM - RECHNER 1A       * <012>
30 REM *        FUER C64 & C128         * <017>
70 REM ********************************** <120>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";OB
    :PRINT 2,                             <242>
110 FOR X=3 TO OB:FOR Y=2 TO X-1          <240>
120 IF X/Y=INT(X/Y)THEN 140               <103>
130 NEXT Y:PRINT X,                       <088>
140 NEXT X                                <088>
```

Listing 1. »Prim-Rechner 1a«, unser erstes Primzahlenprogramm.

```
10 REM ********************************** <060>
20 REM *        PRIM - RECHNER 1B       * <014>
30 REM *        FUER C64 & C128         * <017>
70 REM ********************************** <120>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";OB
    :PRINT 2,                             <242>
110 FOR X=3 TO OB:FOR Y=2 TO X/2          <244>
120 IF X/Y=INT(X/Y)THEN 140               <103>
130 NEXT Y:PRINT X,                       <088>
140 NEXT X                                <088>
```

Listing 2. »Prim-Rechner 1b«, diese Variante von Listing 1 läuft anstelle von x-1 nur noch bis x/2

Erste Verbesserungen

Das geänderte Programm »Prim-Rechner 1b« finden Sie unter Listing 2.

Das Programm ist identisch mit dem Prim-Rechner 1a bis auf Zeile 110. Die Y-Schleife läuft statt bis X-1 jetzt nur noch bis X/2.

Welche Vorteile uns dies gebracht hat, können Sie aus der Zeitentabelle 1 entnehmen. Es ist erstaunlich, aber die Zeiten haben sich nahezu halbiert.

Wir wollen uns aber damit noch nicht zufriedengeben. Wenn Sie sich daran erinnern, wie lange das erste Programm benötigt, um die Primzahlen bis 740914799 zu berechnen, so hilft uns unser zweites Programm auch noch nicht viel weiter.

Deshalb werden wir uns gleich an eine erneute Verbesserung machen. Wie Sie vielleicht beim obigen Durchspielen des Programms mit den Werten 21 und 23 bemerkt haben, wurde die 22 diskret übergangen. Der Grund dafür wird Ihnen sicher auch schon klar sein: Alle geraden Zahlen sind keine Primzahlen, da sie ja durch 2 teilbar sind. Also können wir an unsere X-Schleife ein STEP2 anhängen, damit alle geraden Zahlen übergangen werden. Diese Ein-

sparung bringt aber keine große Zeitersparnis, weil die geraden Zahlen in der Y-Schleife sowieso als erste erkannt worden wären. Deshalb machen wir uns weitere Gedanken. Dazu müssen wir einen kleinen Exkurs in die Mathematik machen.

Wenn Sie eine Nicht-Primzahl in zwei Faktoren zerlegen, so liegt immer ein Faktor über der Wurzel der Nicht-Primzahl und einer darunter. Bei Quadratzahlen liegen beide Faktoren natürlich auf der Wurzel.

Zerlegt man die Zahl in mehrere Faktoren, kann es sein, daß alle Faktoren kleiner sind als die Wurzel. Aber es wird Ihnen nie gelingen, eine Zahl so zu zerlegen, daß kein einziger Faktor kleiner ist als die Wurzel. Dieses wollen wir uns nun zunutze machen. Auf eine Primzahl bezogen heißt das: Hat eine Zahl keinen Teiler, der kleiner oder gleich der Wurzel ist, so hat die Zahl gar keinen Teiler und ist somit eine Primzahl. Die Schlußfolgerung für unser Programm ist, daß wir die Y-Schleife nicht mehr bis X/2 laufen lassen müssen, sondern nur noch bis zur Wurzel von X.

Das neue Programm »Prim-Rechner 1c« können Sie Listing 3 entnehmen.

Der Unterschied zwischen Prim-Rechner 1b und 1c besteht lediglich in Zeile 110. An die X-Schleife wurde das

Zeitentabelle 2

angegeben in Sekunden für C128 bzw. C64

PRG. NAME	BIS 50	BIS 100	BIS 200	BIS 500	BIS 1000	BIS 2000	BIS 5000	BIS 10000	BIS 4E05	GRENZE
PR2A		0		2	5		30	64	---	32000
		1		5	10		57	119	---	19000
PR2B		0		2	5		30	62	---	12500
		1		4	10		55	---	---	7500
PR2C		4		23	49					400000
PR2D		8		45	96				---	240000

PR = Prim - Rechner

```
10 REM ********************************     <060>
20 REM *        PRIM - RECHNER 1C     *     <016>
30 REM *        FUER C64 & C128       *     <017>
70 REM ********************************     <120>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";OB
    :PRINT 2,                               <242>
110 FOR X=3 TO OB STEP 2:FOR Y=2 TO SQR(X)  <218>
120 IF X/Y=INT(X/Y)THEN 140                 <103>
130 NEXT Y:PRINT X,                         <088>
140 NEXT X                                  <088>
```

Listing 3. »Prim-Rechner 1c«, eine Berechnungsreduzierung in der Y-Schleife ist sogar bis Wurzel x erreichbar

```
10 REM ********************************     <060>
20 REM *        PRIM - RECHNER 1D     *     <018>
30 REM *        FUER C64 & C128       *     <017>
70 REM ********************************     <120>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";OB
    :PRINT 2,3,                             <008>
110 FOR X=3 TO OB STEP 2:FOR Y=3 TO SQR(X)
    STEP 2                                  <027>
120 IF X/Y=INT(X/Y)THEN 140                 <103>
130 NEXT Y:PRINT X,                         <088>
140 NEXT X                                  <088>
```

Listing 4. Alle geraden Zahlen werden beim »Prim-Rechner 1d« nicht mehr geprüft

STEP2 angefügt, damit alle geraden Zahlen übersprungen werden. Die Y-Schleife läuft statt bis X/2 nur noch bis SQR(X).

Wenn Sie noch einmal einen Blick auf die Zeitentabelle 1 werfen, werden Sie in der Spalte »bis 500« eine Zeitreduzierung auf nur noch ¼ feststellen.

Aber wir wollen uns damit immer noch nicht zufriedengeben, denn an der Y-Schleife lassen sich noch weitere Verbesserungen vornehmen. Der Startwert dieser Schleife ist immer noch 2. Wir haben aber in der X-Schleife schon alle geraden Zahlen ausgeschlossen, und ungerade Zahlen sind sowieso nicht durch 2 teilbar; so können wir die Y-Schleife gleich mit 3 beginnen lassen. Allerdings tritt dabei eine unerwünschte Nebenwirkung auf, nämlich die 3 wird nicht mehr als Primzahl erkannt und wir müssen diese wie die 2 schon vorweg ausgeben.

Da, wie eben schon gesagt, nur ungerade Zahlen in die Y-Schleife gelangen, brauchen wir ja auch nicht mehr zu prüfen, ob diese durch gerade Zahlen teilbar sind, deshalb setzen wir auch hinter die Y-Schleife ein STEP2.

Unser Ergebnis, den Prim-Rechner 1d, sehen Sie in Listing 4.

Gegenüber dem »Prim-Rechner 1c« wurden folgende Veränderungen vorgenommen:

100: Die Primzahl 3 wird schon vorweg wie die 2 ausgegeben.

110: Die Y-Schleife beginnt statt mit 2 jetzt mit 3. Außerdem wurde ein STEP2 an die Y-Schleife angefügt, damit die geraden Zahlen übergangen werden.

Die Zeiten erfahren Sie wieder aus der Tabelle 1.

Sie werden sich jetzt fragen, was denn jetzt noch zu verbessern sei. Nun, man könnte zum Beispiel die X-Schleife erst mit 5 beginnen lassen. Das ist aber alles, was dieses Programm noch hergibt. Man kann auch eine andere Veränderung vornehmen, die aber nicht so wirkungsvoll ist. In Zeile 120 wird zweimal die Rechnung X/Y ausgeführt. Wenn man diese Rechnung nur einmal ausführt und das Ergebnis einer Variablen zuweist, spart dies etwas Zeit, was aber nicht der Rede wert ist, wie Sie in unserer Zeitentabelle erkennen können.

Das entsprechende Programm ist der »Prim-Rechner 1e« (Listing 5).

Gegenüber dem Prim-Rechner 1d wird in Zeile 120 der Variablen Z das Ergebnis X/Y zugeordnet. In der IF-Bedingung wurde das X/Y durch Z ersetzt.

Die Prim-Rechner der Klasse 1 wurden soweit verbessert, daß sich die Zeiten bis auf ein Achtel der Grundzeit von Listing 1 reduziert haben, wenn man die Berechnung bis 200 zugrunde legt. Wir wollen nun ein anderes Verfahren anwenden, mit dem wir noch mehr Zeit sparen werden.

Zeit contra Speicherplatz

Haben Sie schon einmal etwas von dem Sieb des Eratosthenes gehört? Es funktioniert folgendermaßen: Es werden nicht die Primzahlen berechnet, sondern die Nicht-Primzahlen. Folglich sind alle fehlenden Zahlen die von uns gesuchten Primzahlen. Marcus Werner hat ein kleines Programm nach diesem Prinzip entwickelt, das Sie in der 64'er 9/86 auf Seite 65 finden. Ob die Behauptung »Die schnellste Primzahlenberechnung in Basic« wahr ist, werden wir später noch sehen.

Listing 6 (Prim-Rechner 2a) ist ein Programm nach diesem Prinzip.

Die Programmbeschreibung ist hier ein bißchen länger, da das Verfahren nicht ganz so einfach ist:

Allgemein: Es wird ein Variablenfeld P%(x) eingerichtet. Dann werden alle möglichen Produkte gebildet, die also keine Primzahlen sind. Das Feldelement mit dem Produkt

als Index wird auf 1 gesetzt. Gerade Produkte werden gar nicht berechnet und auch nicht auf 1 gesetzt; darauf muß natürlich bei der Ausgabe geachtet werden. Am Ende der Berechnung der Produkte haben alle Feldelemente, deren Index eine Primzahl ist, den Inhalt 0 und können mit einer IF-Bedingung ausgegeben werden.

100: Zuerst wird die Grenze eingegeben und der Variablen »H« zugeordnet. Z ist die Variable, von der dann die Vielfachen gebildet werden, das heißt, zuerst werden alle Vielfachen von 3 auf 1 gesetzt, dann von der nächsten Zahl und so weiter. Z wird im Verlaufe des Programms verändert. T enthält die nach oben abgerundete Wurzel von der Obergrenze H. W gibt an, von wieviel Zahlen die Produkte gebildet werden müssen. G erhält das Quadrat von T, das etwas größer ist als H. G ist also die nächste auf H folgende Quadratzahl. Das Feld P% wird deshalb auch für G Elemente eingerichtet, da dies einfacher zu handhaben ist.

110: Hier findet das eigentliche Berechnen der Nicht-Primzahlen statt. Wenn Sie aber näher hinsehen, stellen Sie fest, daß gar nicht gerechnet wird. Die X-Schleife gibt lediglich an, wie oft die innere Y-Schleife durchlaufen werden soll. Beim ersten Durchlauf der X-Schleife kommen wir mit Z=3 in die Y-Schleife, das heißt, die Y-Schleife beginnt mit 9 und läuft bis G. Mit dem STEP-Befehl wird Y immer um Z*2 erhöht. Also liefert die Y-Schleife die Werte 9, 15, 21, 27, 33 und so weiter. Gerade Zahlen werden durch das *2 übergangen. Alle Feldelemente mit dem Index Y werden auf 1 gesetzt, danach wird Z um 2 erhöht, wiederum um die geraden Zahlen zu übergehen. Also liefert die Y-Schleife im zweiten Durchlauf die Werte 25, 35, 45, 55, 65 etc. Die betreffenden Feldelemente werden wie oben schon beschrieben gesetzt. Falls Sie sich die Frage stellen, warum die Y-Schleife erst mit dem Quadrat von Z beginnt, so lautet die Antwort, daß die Elemente darunter schon gesetzt sind. Hätte die Y-Schleife im zweiten Fall mit 15 begonnen, so wäre dies überflüssig gewesen, da die erste Y-Schleife für Z=3 die 15 schon als Nicht-Primzahl gesetzt hat.

120: Diese Zeile dient ausschließlich der Ausgabe der Primzahlen. Die 2 wird vorweg ausgegeben. Die weiteren Primzahlen werden durch die X-Schleife bestimmt, wobei die geraden Indices gleich wieder übergangen werden. In der IF-Bedingung wird geprüft, ob der Inhalt des Feldelements 0 ist, gegebenenfalls erfolgt die Ausgabe auf dem Bildschirm.

130: Das NEXT gehört zur X-Schleife der Ausgabe. Es mußte lediglich auf die nächste Zeile gezogen werden, da Zeile 120 mit einer IF-Bedingung endet.

Nach dieser ausführlichen Programmbeschreibung hoffe ich, daß Sie alle einen Einblick in dieses fast nicht mehr zu verbessernde Programm erhalten haben.

Aber das Beste an diesem Programm können Sie der Zeitentabelle 2 entnehmen.

Die Bearbeitungsgeschwindigkeit dieses Programms ist einfach hervorragend. Dafür hat dieses Programm einen großen Nachteil. Wenn Sie die Primzahlen bis 10000 berechnen lassen wollen, brauchen Sie ein Feld mit 10000 Elementen. Da jedes Integer-Element 2 Byte benötigt, können Sie sich schon ausrechnen, daß Sie nicht allzu weit mit diesem Programm kommen.

Ein paar Anmerkungen zur Zeitentabelle 2: Da das Programm so schnell ist, hat es sich nicht gelohnt, die Zeiten für die Berechnung der Primzahlen bis 50, 200 und 2000 zu ermitteln. In der letzten Spalte ist angegeben, bis zu welcher Zahl Sie die Primzahlen berechnen lassen können.

```
10 REM *********************************         <060>
20 REM *         PRIM - RECHNER 1E       *         <020>
30 REM *         FUER C64 & C128         *         <017>
70 REM *********************************         <120>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";OB
    :PRINT 2,3,                                   <008>
110 FOR X=3 TO OB STEP 2:FOR Y=3 TO SQR(X)
    STEP 2                                        <027>
120 Z=X/Y:IF Z=INT(Z)THEN 140                     <028>
130 NEXT Y:PRINT X,                               <088>
140 NEXT X                                        <088>
```

Listing 5. »Prim-Rechner 1e«, die Berechnung x/y in Zeile 120 wird einer Variablen zugewiesen, ein erneuter Zeitgewinn

```
10 REM *********************************         <060>
20 REM *       PRIM - RECHNER 2B         *         <015>
30 REM *       FUER C64 & C128           *         <017>
70 REM *********************************         <120>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";H:
    Z=3:T=INT(SQR(H)+1):W=(T-1)/2:G=T*T:DI
    M P(G)                                        <250>
110 FOR X=1 TO W:FOR Y=Z*Z TO G STEP Z*2:P
    (Y)=1:NEXT:Z=Z+2:NEXT                         <238>
120 PRINT 2,:FOR X=3 TO H STEP 2:IF P(X)=.
    THEN PRINT X,                                 <194>
130 NEXT                                          <140>
```

Listing 7. »Prim-Rechner 2b«, in dieser Version des Listings 5 ist der Variablentyp (reale Zahl/Integer-Wert) entscheidend

```
10 REM *********************************         <060>
20 REM *         PRIM - RECHNER 2A       *         <013>
30 REM *         FUER C64 & C128         *         <017>
40 REM * ------------------------------ *         <045>
50 REM *   BY:   THOMAS HANSCH           *         <237>
60 REM *         DELMENHORST (1986)      *         <175>
70 REM * NACH:  MARCUS WERNER            *         <083>
80 REM *         64'ER 9/86              *         <028>
90 REM *********************************         <140>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";H:
    Z=3:T=INT(SQR(H)+1):W=(T-1)/2:G=T*T:DI
    M P%(G)                                       <057>
110 FOR X=1 TO W:FOR Y=Z*Z TO G STEP Z*2:P
    %(Y)=1:NEXT:Z=Z+2:NEXT                        <221>
120 PRINT 2,:FOR X=3 TO H STEP 2:IF P%(X)=
    .THEN PRINT X,                                <094>
130 NEXT                                          <140>
```

Listing 6. »Prim-Rechner 2a« arbeitet nach dem Prinzip des Eratosthenes

```
10 REM *********************************         <A99>
20 REM *       PRIM - RECHNER 2C         *         <AHA>
30 REM *       NUR FUER C128             *         <APB>
40 REM * ------------------------------ *         <B1C>
50 REM *  BY:  THOMAS HANSCH             *         <D9S>
60 REM *       DELMENHORST (1986)        *         <BHE>
70 REM *********************************         <BPF>
80 :                                              <NAT>
90 :                                              <NQV>
100 DEF FN AD(Y)=INT((Y-1)/16)+10000              <16K>
110 DEF FN FA(Y)=2↑(((Y-1)/16-INT((Y-1)/16))*8)   <414>
120 INPUT "{CLR,2DOWN,2SPACE}OBERGRENZE";H: Z=3:
    T=INT(SQR(H)+1): W=(T-1)/2: G=T*T             <QRO>
130 FOR X=10000 TO 10000+H/16: POKE X,0: NEXT     <230>
140 FOR X=1 TO W: FOR Y=Z*Z TO G STEP Z*2: BANK 0
    : POKE FN AD(Y),PEEK(FN AD(Y)) OR FN FA(Y): N
    EXT : Z=Z+2: NEXT                             <CBA>
150 PRINT 2,: FOR X=3 TO H STEP 2: BANK 0: IF(PEE
    K(FN AD(X)) AND FN FA(X))=0 THEN PRINT X,     <UAA>
160 NEXT                                          <180>
```

Listing 8. Der Zustand Primzahl/keine Primzahl wird bei »Prim-Rechner 2c« in Bits gespeichert. Das Programm läuft nur auf dem C 128. Bitte Eingabehinweise beachten

Um nun noch einmal auf die Behauptung vom schnellsten Basic-Programm zur Berechnung von Primzahlen zurückzukommen – es muß widersprochen werden. Ersetzt man die Integer-Variable P% durch die reelle Variable P, so kann man noch ein paar Sekunden sparen, denn bevor eine Operation mit einer Integer-Zahl ausgeführt wird, muß sie erst in eine reelle Zahl umgewandelt werden. Soll das Ergebnis wieder einer Integer-Variablen zugeordnet werden, so erfolgt wieder eine Umrechnung. Diese Umrechnungen kann man sich sparen, indem man, wie oben schon gesagt, die Integer-Variable durch die reelle Variable ersetzt. Das Programm »Prim-Rechner 2b« können Sie Listing 7 entnehmen.

Es ist im Gegensatz zum Prim-Rechner 2a nur die Variable P% duch P ersetzt worden.

Der Haken an dieser Änderung ist eine noch größere Platzverschwendung, denn eine reelle Zahl benötigt 5 Byte Speicherplatz im Gegensatz zu den Integer-Zahlen mit 2 Byte.

Theoretisch benötigen wir ja gar keine ganze Zahl, um den Zustand prim oder nichtprim zu unterscheiden.

Dafür reicht ein einziges Bit aus, und da die geraden Zahlen sowieso nicht mehr gesetzt, sondern vom Programm übergangen werden, kann man gleich die Speicherplätze dafür einsparen.

Das Problem ist bloß die richtige Adressierung des richtigen Bits. Eine derartige Lösung könnte folgendermaßen aussehen:

Für den C 128: Prim-Rechner 2c (Listing 8)
Für den C 64: Prim-Rechner 2d (Listing 9)
Programmbeschreibung:

100: Definition einer Funktion AD, die in Abhängigkeit von der Zahl die Adresse des betreffenden Byte ermittelt.

110: Definition einer Funktion FA, die in Abhängigkeit von der Zahl den Wert des betreffenden Bits liefert. Ein Bit wird mit OR FN FA(x) gesetzt.

120: entspricht der Zeile 100 vom Prim-Rechner 2a.

130: Löschen aller benötigten Bits.

140: entspricht der Zeile 110 vom Prim-Rechner 2a. Allerdings wird das Setzen des Bit durch POKE erledigt.

150/160: Ausgabe der Primzahlen entsprechend Zeile 120 des Prim-Rechners 2a.

Die Versionen für den C 128 und C 64 sind unterschiedlich, weil dem C 128 durch BANK0 noch zusätzlich der Speicherbereich angegeben werden muß, auf den sich die PEEK's und POKE's beziehen.

Zeitentabelle 3

angegeben in Sekunden für C128 bzw. C64

PRG. NAME	BIS 50	BIS 100	BIS 200	BIS 500	BIS 1000	BIS 2000	BIS 5000	BIS 10000	BIS 4E05	GRENZE
PR3		4 5		13 20	25 44		153 281	349 644	10 H	1E09 1E09

PR = Prim – Rechner H = Stunden

```
10 REM ********************************   <060>
20 REM *        PRIM - RECHNER 2D     *   <019>
30 REM *        NUR FUER C64          *   <167>
70 REM ********************************   <120>
100 DEF FN AD(Y)=INT((Y-1)/16)+10000     <165>
110 DEF FN FA(Y)=2↑(((Y-1)/16-INT((Y-1)/16
    ))*8)                                <168>
120 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";H:
    Z=3:T=INT(SQR(H)+1):W=(T-1)/2:G=T*T  <218>
130 FOR X=10000 TO 10000+H/16:POKE X,0:NEX
    T                                    <083>
140 FOR X=1 TO W:FOR Y=Z*Z TO G STEP Z*2:P
    OKE FN AD(Y),PEEK(FN AD(Y))OR FN FA(Y)
    :NEXT:Z=Z+2:NEXT                     <135>
150 PRINT 2,:FOR X=3 TO H STEP 2:IF(PEEK(F
    N AD(X))AND FN FA(X))=0 THEN PRINT X, <084>
160 NEXT                                 <170>
```

Listing 9. »Prim-Rechner 2d« ist wie Listing 8, jedoch für Basic 2.0 auf dem C 64

```
10 REM ********************************   <060>
20 REM *        PRIM - RECHNER 3      *   <056>
30 REM *        FUER C64 & C128       *   <017>
70 REM ********************************   <120>
100 DIM PR(4000)                         <131>
110 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE ";O
    B                                    <010>
120 PR(1)=2:PR(2)=2:PZ=2:MX=1:W=4:PRINT,1;
    2,                                   <085>
130 FOR Z=3 TO OB STEP 2:IF Z>W THEN MX=MX
    +1:W=PR(MX)*PR(MX)                   <098>
140 FOR ZA=2 TO MX:F=Z/PR(ZA):IF F<>INT(F)
    THEN NEXT ZA:PRINT PZ;Z,:PZ=PZ+1:IF PZ
    <4000 THEN PR(PZ-1)=Z                <232>
150 NEXT Z                               <114>
```

Listing 10. Zur schnelleren Berechnung großer Primzahlen werden werden beim »Prim-Rechner 3« die ersten 4000 Primzahlen in einem Array gespeichert

Erneute Problemanalyse

Durch diese platzsparende Methode können wir beim C 128 die Primzahlen bis etwa 400000, bei C 64 bis zirka 240000 berechnen lassen. Allerdings kostet die Berechnung der Adressen so viel Zeit, daß es nicht sinnvoll ist, das Programm so weit laufen zu lassen. Die Zeiten können Sie in der Zeitentabelle nachschauen.

Wenn wir alles Bisherige zusammenfassen, haben wir einen Prim-Rechner (1e), der Primzahlen bis zu einer Milliarde berechnen könnte, er ist aber viel zu langsam. Der Prim-Rechner 2a, den wir kennengelernt haben, wäre wahrscheinlich schnell genug, er benötigt aber zuviel Speicherplatz, so daß wir lediglich Primzahlen bis 32000 (beziehungsweise 19000) berechnen lassen können. Wir wollen nun versuchen, die Vorteile beider Programme »unter einen Hut« zu bringen.

Damit wir auf ein wirklich neues Prinzip kommen, schauen wir uns die Beschreibung einer Primzahl aus dem Lexikon noch einmal genauestens an:

Primzahl, in der Mathematik eine natürliche Zahl, die keine andere natürliche Zahl als Teiler hat, außer 1 und sich selbst. Die ersten Primzahlen sind 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, ... Jede natürliche Zahl ist entweder prim oder kann als Produkt von Primzahlen dargestellt werden, zum Beispiel 780 = 2 * 2 * 3 * 5 * 13.

Es gibt unendlich viele Primzahlen, obwohl sie mit wachsendem absoluten Betrag immer spärlicher werden.

Bei unserem ersten Programm haben wir sozusagen nur vom ersten Satz Gebrauch gemacht, denn wir haben die zu untersuchende Zahl einfach durch verschiedene natürliche Zahlen geteilt. War das Ergebnis ebenfalls eine natürliche Zahl, so ist die untersuchte Zahl keine Primzahl. Unser zweiter Prim-Rechner hat vom zweiten Satz Gebrauch gemacht, indem wir einfach Produkte gebildet haben und dann die übriggebliebenen Zahlen als Primzahlen ausgegeben haben. Bei diesem Verfahren kam es aber vor, daß bestimmte Zahlen mehrmals als Produkte vorkommen.

Der zweite Satz der Beschreibung aus dem Lexikon läßt sich auch für Nicht-Primzahlen umformulieren:

Eine Zahl ist keine Primzahl, wenn sie als Produkt von Primzahlen dargestellt werden kann.

Vielleicht sehen Sie jetzt schon, worauf es hinausläuft:

Beim Prim-Rechner 1e haben wir die zu untersuchende Zahl durch alle ungeraden Zahlen, mit 3 beginnend, bis zur Wurzel der Zahl geteilt. Unter diesen ungeraden Zahlen, durch die geteilt wird, sind sicherlich einige Nicht-Primzahlen enthalten, wie zum Beispiel 9, 15 und 21. Da die Primzahlen in höheren Bereichen immer spärlicher werden, werden die ungeraden Nicht-Primzahlen immer mehr.

Um nun das Prüfen einer Zahl zu optimieren, muß also die zu untersuchende Zahl nur durch Primzahlen geteilt

```
10 REM ********************************   <060>
20 REM *        PRIM - ZWILLING 1      *   <137>
30 REM *        FUER C64 & C128        *   <017>
70 REM ********************************   <120>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";OB
    :PRINT,2;3,3;5,5;7,11;13,              <167>
110 FOR X=17 TO OB STEP 10                 <171>
120 T=X+2:GOSUB 190:IF FLAG=0 THEN F2=0:GO
    TO 140                                 <221>
130 F2=1:T=X:GOSUB 190:IF FLAG=1 THEN PRIN
    T X;X+2,                               <123>
140 T=X+4:GOSUB 190:IF FLAG=0 THEN 170     <052>
150 IF F2=1 THEN PRINT X+2;X+4,            <192>
160 T=X+6:GOSUB 190:IF FLAG=1 THEN PRINT X
    +4;X+6,                                <171>
170 NEXT                                   <180>
180 END                                    <182>
190 FOR Y=3 TO SQR(T)STEP 2                <137>
200 IF T/Y=INT(T/Y)THEN FLAG=0:GOTO 220    <115>
210 NEXT Y:FLAG=1                          <052>
220 RETURN                                 <022>
```

Listing 11. »Prim-Zwilling 1«, unser erstes Programm zur Berechnung von Primzahlenzwillingen

werden. Diese Primzahlen müssen ständig vorhanden sein. Wenn Sie an den Prim-Rechner 2 denken, so haben Sie auch schon die richtige Speichermöglichkeit gefunden.

Sicher werden Sie denken, daß nicht alle Primzahlen bis zu 1 Milliarde im Computer gespeichert werden können. Da haben Sie schon recht, aber wir benötigen ja nur die Primzahlen bis zur Wurzel von 1 Milliarde, das heißt bis 31623. Da wir unsere Primzahlen aber nicht durch Produktbildung, sondern durch das Teilverfahren vom Prim-Rechner 1 bestimmen wollen, werden in einem Feld alle ermittelten Primzahlen gespeichert. Das Feld könnten wir vom Typ Integer einrichten, da die Zahl 31623 kleiner als 32767, der größten im C64 und C128 darstellbaren Integer-Zahl, ist. Aber wir haben selbst beim C64 genügend Platz, eine reelle Variable einzurichten. Sicherheitshalber kann die Anzahl der Elemente auch auf 4000 heraufgesetzt werden, obwohl die Zahl 31627 (die nächstfolgende Primzahl auf 31623) erst zirka die 3350. Primzahl ist. Der Index eines Feldelements gibt zudem noch an, um die wievielte Primzahl es sich handelt.

Nach diesen ganzen Vorüberlegungen erhalten Sie ein entsprechendes Listing, den »Prim-Rechner 3« (Listing 10). Hier nun die Programmbeschreibung:

100: Einrichten des Feldes PR für 4000 Elemente.

110: Eingabe der Obergrenze.

120: Die Zahl 2 wird als erste Primzahl (Index von PR ist 1) gespeichert. Dem Feldelement PR(2) muß schon ein Wert vorgegeben werden, damit es in Zeile 140 nicht zum Abbruch kommt. PZ gibt an, die wievielte Primzahl gerade berechnet wird. MX ist der maximale Index des Feldelements, dessen Primzahl die letzte ist, durch die die zu prüfende Zahl geteilt wird. W enthält die Zahl, bis zu der der maximale Index MX nicht erhöht werden muß. Danach folgt die Ausgabe der 2 als erste Primzahl. Da wir in diesem Programm einen Zähler PZ haben, der die Primzahlen zählt, wird diese Anzahl bei der Ausgabe mit ausgegeben.

130: Die Z-Schleife läuft von 3 bis zur Obergrenze in 2er-Schritten, damit die geraden Zahlen übergangen werden. Wird Z größer W, erhöht sich der maximale Index um 1 und W wird neu berechnet.

140: In der ZA-Schleife wird überprüft, wo es sich um eine Primzahl handelt. ZA ist der Index von PR. Er beginnt aber erst mit 2, da gerade Zahlen nicht in die ZA-Schleife kommen. Folglich braucht Z durch 2 nicht mehr geteilt zu werden. ZA läuft bis zum maximalen Index MX durch.

Es wird jeweils der Quotient F=Z/PR(ZA) gebildet, das heißt, F wird das Ergebnis aus der zu prüfenden Zahl geteilt durch die Primzahlen zugeordnet. Hat F keine Nachkommastellen, so wird das Programm in der nächsten Zeile fortgesetzt. Die Zahl Z ist dann keine Primzahl. Hat F bei jeder Berechnung Nachkommastellen, so wird die Primzahl mit ihrem Index ausgegeben. Der Index PZ wird daraufhin um 1 erhöht. Solange der Index noch kleiner 4000 ist, wird die Primzahl Z in PR gespeichert.

150: Das NEXT leitet die Prüfung der nächsten Zahl ein.

Damit wir auch sehen, ob das Programm unseren Erfolgserwartungen gerecht wird, finden Sie die entsprechenden Angaben in Zeitentabelle 3.

Vielleicht sind Sie etwas enttäuscht, da die Zeiten für 100 und 500 Primzahlen ungefähr mit denen vom Prim-Rechner 1e gleich sind. Eine Zeiteinsparung läßt sich allerdings erst ab 1000 erkennen. Der entscheidende Vorteil liegt darin, daß die Zeiten nur gering ansteigen. So errechnet der Prim-Rechner die Primzahlen bis 400000 in zirka 10 Stunden. Würden Sie dieses Programm einsetzen, um 740914799 auf eine Primzahl testen zu lassen, so können wir ja mal eine kleine Hochrechnung anstellen:

Vermutlich wird das Programm im Durchschnitt 30 Minuten für ein 10000er-Intervall benötigen. Das entspricht 37000 Stunden oder guten 4 Jahren. Erinnern Sie sich noch, wie lange unser Prim-Rechner 1 dafür benötigte?

Sicher läßt sich auch unser Prim-Rechner 3 noch etwas beschleunigen, indem man vielleicht die ersten Primzahlen durch die Produktbildung berechnet. Wenn Sie Spaß daran haben, können Sie ja vielleicht noch einen Prim-Rechner entwickeln, der nur noch 1 Jahr benötigt, um alle Primzahlen bis 740914799 zu berechnen.

Oder vielleicht möchten Sie ja anhand einer neuen Problemstellung Ihre eigenen Verbesserungen und eigenen Prinzipien entwickeln, deshalb hier eine neue Aufgabe:

Schreiben Sie ein Programm, das alle Primzahlenzwillinge berechnet.

Primzahlenzwillinge sind benachbarte Primzahlen, deren Differenz 2 ist, wie zum Beispiel 3/5, 5/7, 11/13, 17/19 und 29/31.

Die Zahl 2 spielt wieder eine Ausnahmerolle, denn das Paar 2/3 ist auch ein Primzahlenzwilling. Sie könnten es sich sehr einfach machen, wenn Sie den Prim-Rechner 2 verwenden und bloß die Ausgabe ändern. Wenn Ihr Programm die Zwillinge bis zu einer Größe von 1 Milliarde berechnen kann, so wäre das aber viel besser.

Es braucht wohl nicht extra erwähnt zu werden, daß das Programm so schnell wie möglich arbeiten soll.

Zu schwierig? Betrachten wir also die Primzahlenzwillinge etwas genauer.

Ungleiche Zwillinge

Der einfachste Weg, einen Primzahlenzwilling zu finden, ist, eine Primzahl zu ermitteln und dann zu prüfen, ob die um 2 größere Zahl auch eine Primzahl ist. Von daher ist es am einfachsten, einen der bis jetzt entwickelten Prim-Rechner zu verwenden. Daraus läßt sich folgern, daß die Zeit, die das betreffende Programm benötigt, verlängert wird, da zu der Ermittlung einer Primzahl noch zusätzlich

geprüft werden muß, ob die um 2 größere Zahl auch eine Primzahl ist. Ein Beispiel für diese Vorgehensweise wird Ihnen später vorgestellt.

Zuerst erhalten Sie ein Programm, das auf den Prim-Rechner 1e aufbaut, aber schneller ist. Sie haben richtig gelesen, das Programm ist wirklich schneller. Der Grund dafür ist, Sie werden es sich schon denken, ein völlig neues Prinzip. Und zwar stellt man fest, wenn man beispielsweise das 10er-Intervall von 15 bis 25 untersucht, daß nur die Primzahlenpaare mit den Endziffern 7/9, 9/1 und 1/3 möglich sind. Paare mit der Endziffer 5 sind nicht möglich, da die betreffende Zahl ja immer durch 5 teilbar ist. Der Bereich bis zur Zahl 15 bildet hier dann eine Ausnahme, das heißt, die Zwillinge werden »von Hand« ausgegeben. Der entscheidende Vorteil ist, daß immer ein 10er-Intervall untersucht wird. Es werden aber nicht alle vier betreffenden Endziffern geprüft, sondern die Prüfung beginnt mit der Endziffer 9. Ist dies keine Primzahl, braucht die Endziffer 7 gar nicht mehr geprüft zu werden, da das Paar 7/9 schon nicht mehr möglich ist. War die Zahl mit der Endziffer 9 eine Primzahl, so muß auch Endziffer 7 geprüft werden. Handelt es sich dabei auch um eine Primzahl, so wird das Paar ausgegeben. Als nächstes wird die Endziffer 1 überprüft. Ist es keine Primzahl, so ist die Untersuchung des Intervalls schon beendet. Ist Endziffer 1 eine Primzahl, und war Endziffer 9 eine Primzahl, so wird erst einmal das Paar 9/1 ausgegeben. Ansonsten wird die Endziffer 3 noch geprüft und entsprechend das Paar 1/3 ausgegeben oder auch nicht, wenn Endziffer 3 keine Primzahl ist. Die Prüfung der einzelnen Zahlen erfolgt wie im Prim-Rechner 1e.

Da Sie jetzt das Verfahren kennen, hier die Programmbeschreibung des etwas längeren Listings 11, dem »Prim-Zwilling 1«.

100: Eingabe der Obergrenze und Ausgabe der Primzahlenzwillinge bis 15.

110: X-Schleife beginnend mit 17 bis zur Obergrenze in 10er-Schritten.

120: T erhält die Zahl, die auf eine Primzahl geprüft werden soll, danach erfolgt der Aufruf des Unterprogramms in Zeile 190. Die Variable FLAG ist gesetzt, wenn es sich um eine Primzahl handelte. Die Variable F2 ist ein Hilfs-Flag und wird gelöscht, wenn Endziffer 9 keine Primzahl war. Danach wird nach 140 gesprungen.

130: War Endziffer 9 eine Primzahl, so wird F2 auf 1 gesetzt. Danach wird Endziffer 7 geprüft. Ist FLAG gesetzt, so wird das Paar 7/9 ausgegeben.

140: Endziffer 1 wird geprüft. Ist es keine Primzahl, so wird gleich nach 170 gesprungen zur Untersuchung des nächsten Intervalls.

150: War die Endziffer 9 eine Primzahl, so wird das Paar 9/1 ausgegeben.

160: Danach wird Endziffer 3 geprüft. Ist es auch eine Primzahl, so wird auch das Paar 1/3 ausgegeben.

170: Fortsetzung der X-Schleife.

180: Ende des Programms.

190: Beginn des Unterprogramms mit einer Y-Schleife, die von 3 bis zur Wurzel der Zahl T in 2er-Schritten läuft.

200: Treten bei der Division einmal keine Nachkommastellen auf, so wird FLAG gelöscht und es erfolgt ein Sprung nach 220.

210: Ansonsten wird die Y-Schleife fortgesetzt und am Ende FLAG gesetzt.

220: Ende des Unterprogramms; Rücksprung ins Hauptprogramm.

Die Ergebnisse können Sie der Zeitentabelle 4 entnehmen.

Wenn Sie die Zeiten vom Prim-Zwilling 1 mit dem Prim-Rechner 1e vergleichen, sehen Sie, daß bei der Berechnung bis 1000 ganze 10 (beziehungsweise 21 bei C 64) Sekunden eingespart wurden.

Zeitentabelle 4

angegeben in Sekunden für C128 bzw. C64

PRG. NAME	BIS 50	BIS 100	BIS 200	BIS 500	BIS 1000	BIS 2000	BIS 5000	BIS 10000	BIS 4E05	GRENZE
PZ1		1 3		9 18	21 39		149 278			1E09 1E09
PZ2		0 1		3 6	6 13		36 69	75 143		32000 19000

PZ = Prim - Zwilling

Diesen Prim-Zwilling 1 können Sie, wenn Sie die Zeit haben, bis zu einer Milliarde rechnen lassen. Falls Sie aber die Primzwillinge nur bis 32000 (beziehungsweise 19000) benötigen, können sie sich viel Arbeit sparen.

Der Prim-Rechner 2a eignet sich nämlich auch hervorragend für die Berechnung von Primzahlenzwillingen. Es braucht lediglich die Zeile für die Ausgabe geändert zu werden, denn es muß geprüft werden, ob die um 2 größere Zahl auch eine Primzahl ist. Wie das im einzelnen aussieht, sehen Sie im »Prim-Zwilling 2« (Listing 12).

Zur Funktionsweise: Das Programm entspricht dem Prim-Rechner 2a, nur Zeile 120 ist verändert.

120: Zuerst wird der Ausnahme-Zwilling 2/3 ausgegeben. Danach beginnt die X-Schleife von 3 bis H in 2er-Schritten zum Prüfen von Primzahlenzwillingen. Ist das Feldelement mit dem Index X und mit dem Index X+2 nicht gesetzt, so sind beide Zahlen Primzahlen, folglich handelt es sich um einen Zwilling und er wird ausgegeben.

Die Zeiten des Prim-Zwillings 2 sind natürlich länger als die des Prim-Rechner 2a, da bei der Ausgabe einige Über-

Zeitentabelle 5 (Zusammenfassung)

angegeben in Sekunden für C128 bzw. C64

PRG. NAME	BIS 50	BIS 100	BIS 200	BIS 500	BIS 1000	BIS 2000	BIS 5000	BIS 10000	BIS 4E05	GRENZE
PR1A	2 5	8 16	33 63							
PR1B	1 3	5 9	18 34	87 165						
PR1C	1 2	2 5	7 13	21 40	49 95					
PR1D	1 2	2 4	5 9	14 28	33 63	76 147				
PR1E	1 2	2 4	4 9	14 26	31 60	72 137				
PR2A		0 1		2 5	5 10		30 57	64 119	--- ---	32000 19000
PR2B		0 1		2 4	5 10		30 55	62 ---	--- ---	12500 7500
PR2C PR2D		4 8		23 45	49 96				 ---	400000 240000
PR3		4 5		13 20	25 44		153 281	349 644	10 H	1E09 1E09
PZ1		1 3		9 18	21 39		149 278			1E09 1E09
PZ2		0 1		3 6	6 13		36 69	75 143		32000 19000

PR = Prim - Rechner PZ = Prim - Zwilling

prüfungen mehr gemacht werden müssen. Aber dennoch ist er erheblich schneller als Prim-Zwilling 1.

Die beiden vorgestellten Prim-Zwillinge basieren auf den Prim-Rechnern 1 beziehungsweise 2. Sicherlich kann man auch einen Prim-Zwilling mit dem Prim-Rechner 3 als Grundlage erstellen. Der würde allerdings viel langsamer werden. Bis zur 4000sten Primzahl ist es noch relativ einfach, da bis dahin die Primzahlen gespeichert werden und man nur die Differenz zur letzten Primzahl überprüfen muß. Allerdings wird es schon schwieriger, wenn man über die 4000ste Primzahl hinauskommt, da man dann die letzte Primzahl zwischenspeichern müßte, um sie mit der nächsten zu vergleichen. Diese ganzen Maßnahmen nehmen einfach zuviel Zeit in Anspruch. Wir werden aber sehen, wofür sich der Prim-Rechner 3 besser eignet.

Damit wären unsere Programme zur Berechnung aller Primzahlen abgeschlossen, deshalb erhalten Sie als Zusammenfassung noch einmal die Zeiten aller bisher entwickelten Programme auf einen Blick in Zeitentabelle 5 (Zusammenfassung).

Prim oder nichtprim?

Der nächste und letzte Teil dieses Kurses wird sich mit der Prüfung einzelner Zahlen auf den Zustand prim oder nichtprim beschäftigen.

Nachdem wir die Berechnung von allen Primzahlen bis zu einer gewissen Obergrenze abgeschlossen haben, werden wir diesmal einzelne Zahlen auf den Zustand prim oder nichtprim testen lassen.

Für diese Zwecke lassen sich natürlich auch wieder unsere Prim-Rechner einsetzen. Die Frage ist nur, welche Prim-Rechner geeignet sind. Da der Prim-Rechner 2 nur

```
10 REM ********************************      <060>
20 REM *        PRIM - ZWILLING 2       *    <139>
30 REM *        FUER C64 & C128         *    <017>
70 REM ********************************      <120>
100 INPUT"{CLR,2DOWN,2SPACE}OBERGRENZE";H:
    Z=3:T=INT(SQR(H)+1):W=(T-1)/2:G=T*T:DI
    M P%(G)                                  <057>
110 FOR X=1 TO W:FOR Y=Z*Z TO G STEP Z*2:P
    %(Y)=1:NEXT:Z=Z+2:NEXT                   <221>
120 PRINT,2;3,:FOR X=3 TO H STEP 2:IF P%(X
    )=.AND P%(X+2)=.THEN PRINT X;X+2,        <231>
130 NEXT                                     <140>
```

Listing 12. Fast wie »Prim-Rechner 2a« arbeitet auch »Prim-Zwilling 2«

```
10 REM ********************************      <060>
20 REM *        PRIM - GENERATOR        *    <223>
30 REM *        FUER C64 & C128         *    <017>
70 REM ********************************      <120>
100 PRINT"{CLR,2DOWN,2SPACE}DIE PRIMZAHLEN
    -DATEI WIRD ERSTELLT !"                  <000>
110 DIM PR(4000)                             <141>
120 OPEN 2,8,2,"@0:PRIMZAHLEN,S,W"           <193>
130 PR(1)=2:PZ=2:W=4:PRINT#2,2               <253>
140 FOR Z=3 TO 33500 STEP 2                  <117>
150 IF Z>=W THEN MX=MX+1:W=PR(MX)*PR(MX)     <134>
160 FOR ZA=1 TO MX-1:F=Z/PR(ZA):IF F<>INT(
    F)THEN NEXT ZA:PRINT#2,Z:PR(PZ)=Z:PZ=P
    Z+1                                      <224>
170 NEXT Z                                   <134>
180 CLOSE 2                                  <199>
```

Listing 13. Der »Prim-Generator« richtet eine sequentielle Datei ein, die etwa 4000 Primzahlen enthält

bis 32000 (beziehungsweise 19000) verwendbar ist, scheidet er schon einmal aus. Der entscheidende Unterschied zwischen den zwei verbleibenden Prim-Rechnern ist, daß die Zeiten beim Prim-Rechner 1 stärker ansteigen, was wiederum bedeutet, daß die Prüfung größerer Zahlen, relativ gesehen, länger dauert als beim Prim-Rechner 3. Von daher werden wir uns für den Prim-Rechner 3 entscheiden.

Dabei tritt allerdings wieder ein Problem auf; der auserwählte Prim-Rechner benötigt etwa die ersten 4000 Primzahlen, um überhaupt eine Zahl testen zu können. Hierfür bieten sich wieder zwei Lösungsmöglichkeiten an. Erstens kann man die betreffenden Primzahlen in DATA-Zeilen im Programm unterbringen. Das hat zur Folge, daß das Programm 90 Blöcke lang wird. Da die Zahlen nach dem Starten aus den DATA-Zeilen dann in die Feldvariable übergeben werden müssen, steigt der C 64 aus, da ihm der Speicherplatz fehlt. Der C 128 spielt hier aber mit, und deshalb ist auf der Programmdiskette zu dieser Ausgabe auch der entsprechende Prim-Tester 1 (nur für den C 128) enthalten. Um aber den Prim-Rechner 3 auch für den C 64 zu nutzen, muß mit einem Zusatzprogramm eine Datei eingerichtet werden, die die benötigten Primzahlen enthält.

Das »Prim-Generator«-Programm finden Sie unter Listing 13.

Zum Programmaufbau:

100: Information für den Anwender.

110: Einrichten des Feldes PR, vorsichtshalber auf 4000 Elemente.

120: Öffnen der sequentiellen Datei »Primzahlen«.

130: Die 2 wird als erste Primzahl dem Feld zugeordnet, die Hilfsvariablen PZ und W werden gesetzt. Die 2 wird auch als erste Primzahl in die Datei geschrieben.

140: Z-Schleife bis 33500 (weiter werden die Primzahlen nicht benötigt).

150-170: Bekannte Prüfroutine vom Prim-Rechner 3.

180: Schließen der Datei.

Auf Ihrer Diskette müssen für die Datei mindestens noch 109 Blöcke zur Verfügung stehen. Selbstverständlich braucht die Datei nur einmal eingerichtet zu werden. Die Zeit für die Einrichtung beträgt beim C 128 rund 30 Minuten, beim C 64 etwa 50 Minuten.

Alle Vorbereitungen sind abgeschlossen. Wir können uns nun an das Programmieren des Prim-Tester 2 machen.

Wenn Sie sich noch einmal die Programmbeschreibung vom Prim-Rechner 3 ansehen, finden Sie dort die Variable MX, die den Index der Primzahl enthält, die größer ist als die Wurzel der zu untersuchenden Zahl. Diese Variable wird im Programmverlauf immer auf den neuesten Stand gebracht. Da wir die Primzahlen aber nicht von Anfang an berechnen wollen, müssen wir den Index anders berechnen. Da die Primzahlen ja in aufsteigender Reihenfolge vorhanden sind, ist dies mit Hilfe der »binären Suchmethode« kein Problem und geht zudem noch äußerst schnell.

Falls Ihnen diese Methode nicht geläufig sein sollte, hier eine kurze Erklärung: Wird ein bestimmtes Element in einem sortierten Feld benötigt, so nimmt man das mittlere Element und vergleicht, ob es größer oder kleiner ist. Ist der Wert zu groß, wird nur die untere Hälfte weiter betrachtet, ansonsten wird nur die obere Hälfte weiter verwendet. Aus der ermittelten Hälfte nimmt man wieder das mittlere Element und teilt die Hälfte in zwei weitere Hälften, aus denen man sich wieder die heraussucht, in der der gesuchte Wert liegen muß. Verfährt man weiter auf diese Weise, erhält man zwangsläufig den gesuchten Wert.

Als weiteren Service soll der Prim-Tester 2 bei Nicht-Primzahlen sämtliche Primfaktoren ausgeben. Dies läßt sich so realisieren, daß wenn eine Zahl teilbar ist, der Teiler den ersten Primfaktor darstellt. Der Rest wird als neue zu prüfende Zahl verwendet.

Um Sie nun nicht länger auf die Folter zu spannen, hier nun die ausführliche Programmbeschreibung des »Prim-Tester 2« (Listing 14):

100: Einrichten des Feldes für 3584 Elemente (3584 entspricht 2hoch11 + 2hoch10 + 2hoch9). Das erste Element wird nicht benötigt, aber aus Sicherheitsgründen mit der ersten Primzahl belegt.

110: Information für den Anwender.

120: Öffnen der Primzahlen-Datei.

130: Einlesen der Primzahlen und Schließen der Datei.

140-150: Information für den Anwender.

160: Eingabe der zu prüfenden Zahl (EG).

170: Prüfen, ob die Eingabe gültig ist, ansonsten Wiederholung der Eingabe.

180: F ist der Exponent für die binäre Suchmethode. Er wird von 11 bis 0 heruntergezählt. Damit wir hier mit einem ganzen Exponent arbeiten können, wird angenommen, daß das Feld 2hoch12 = 4096 Elemente hat. IN enthält den Index des mittleren Elements sowie X die Wurzel der gesuchten Zahl.

190: Ist F bei 0 angelangt, ist die binäre Suchmethode beendet und das Programm wird in Zeile 230 fortgesetzt.

200: Herabzählen von F.

210: Ist der Inhalt des Elements mit dem Index IN zu klein, so wird die obere Hälfte weiterverwendet, IN wird entsprechend erhöht und es folgt ein Sprung nach 190.

220: Ist der Inhalt zu groß, so wird IN verringert und es erfolgt der Sprung nach 190.

230: Schleife zum Prüfen, ob EG eine Primzahl ist. Läuft die Schleife bis zum Ende, so handelt es sich um eine Primzahl und es erfolgt ein Sprung nach 260.

240: Ist FLAG gelöscht, so läuft das Programm für die Original-Eingabe durch; ist es gesetzt, so ist schon mindestens ein Primfaktor gefunden worden und das Programm läuft jetzt mit dem Rest, um zu prüfen, ob er selbst eine Primzahl ist oder ob er sich in weitere Primzahlen zerlegen läßt. Ist FLAG in diesem Fall 0, so wird ausgegeben, daß EG keine Primzahl ist und FLAG wird gesetzt.

250: Ausgabe eines Primfaktors sowie ein »*« als Multiplikationszeichen. EG wird der Rest der Eingabe, geteilt durch den Primfaktor, zugeordnet. Danach folgt der Sprung nach 180 zum Prüfen des Rests.

260: Ist FLAG gelöscht, so wird ausgegeben, daß EG eine Primzahl ist.

270: Sonst wird einfach die Zahl ausgegeben.

280-300: Frage an den Anwender, ob er noch eine Zahl prüfen lassen will.

310: Ende des Programms.

Die Zahl 2 stellt auch bei diesem Programm eine Ausnahme dar, deshalb wird sie bei der Eingabe nicht akzeptiert. Ein kleines Fehlerchen in diesem Programm ist, daß bei 2er-Potenzen als letzter Primfaktor die 1 ausgegeben wird.

Dies ist mathematisch gesehen falsch, soll uns aber nicht weiter stören, da ein Faktor 1 die Zahl nicht verändert.

Mit diesem Programm sind wir eigentlich am Ziel unserer Träume. Wenn Sie den Prim-Tester 2 nun starten und dann die Zahl 740914799 eintippen, erfahren Sie, ob es sich nun um eine Primzahl handelt oder nicht. Das Ergebnis ist, daß obige Zahl aus den Primfaktoren 22229 und 33331 besteht. Die Zeit, die dieses Programm zur Prüfung einer Zahl benötigt, hängt ganz von der Größe ab. Für kleine Zahlen werden einige Sekunden benötigt, während die Prüfung einer 9stelligen Zahl bis zu über 1 Minute dauern kann.

Um die Zahl 740914799 als Nicht-Primzahl zu entlarven, hätten Sie es sich aber auch einfacher machen können. Mit dem »Prim-Test 740...« (Listing 15), der auf dem Prim-Rechner 1e aufbaut, erhalten Sie nach 78 Sekunden (beziehungsweise 117 Sekunden beim C 64) die betreffenden Primfaktoren.

```
10 REM ********************************     <060>
20 REM *         PRIM - TESTER 2       *    <050>
30 REM *         FUER C64 & C128       *    <017>
70 REM ********************************     <120>
100 DIM PR(3584):PR(0)=2                    <108>
110 PRINT"{CLR,2DOWN,2SPACE}MOMENT BITTE !
     ICH LESE DATEN !"                      <177>
120 OPEN 2,8,2,"PRIMZAHLEN,S,R"             <019>
130 FOR X=1 TO 3584:INPUT#2,PR(X):NEXT:CLO
    SE 2                                    <179>
140 PRINT:PRINT"{CLR,2SPACE}BITTE GEBEN SI
    E NUN DIE ZAHL EIN,"                    <158>
150 PRINT:PRINT"{2SPACE}DIE GEPRUEFT WERDE
    N SOLL:"                                <232>
160 PRINT:INPUT"{2SPACE}";EG:FLAG=0:PRINT   <165>
170 IF EG>=1E+09 OR EG<3 OR EG<>INT(EG)THE
    N PRINT"{4UP}":GOTO 160                 <114>
180 F=11:IN=2↑F:X=SQR(EG)                   <194>
190 IF F=0 THEN 230                         <228>
200 F=F-1                                   <155>
210 IF PR(IN)<X THEN IN=IN+2↑F:GOTO 190     <030>
220 IF PR(IN)>=X THEN IN=IN-2↑F:GOTO 190    <097>
230 FOR X=0 TO IN:F=EG/PR(X):IF F<>INT(F)T
    HEN NEXT:GOTO 260                       <044>
240 IF FLAG=0 THEN PRINT:PRINT EG"IST KEIN
    E PRIMZAHL !":PRINT:PRINT"{2SPACE}PRIM
    FAKTOREN VON"EG":":PRINT:FLAG=1         <066>
250 PRINT PR(X)"*",:EG=EG/PR(X):GOTO 180    <062>
260 IF FLAG=0 THEN PRINT:PRINT"{2SPACE}DIE
     ZAHL"EG"IST EINE PRIMZAHL !"           <227>
270 IF FLAG<>0 THEN PRINT EG                <189>
280 PRINT:PRINT:PRINT"{2SPACE}NOCH EINE BE
    RECHNUNG ? (J/N) "                      <057>
290 GET X$:IF X$="J"THEN GOTO 140           <010>
300 IF X$<>"N"THEN 290                      <173>
310 END                                     <058>
```

Listing 14. »Prim-Tester 2« verwertet die etwa 109 Blocks lang sequentielle Datei, die durch Listing 14 erzeugt wurde

```
10 REM ********************************     <060>
20 REM *    PRIM - TEST 740.914.799    *    <159>
30 REM *        FUER C64 & C128        *    <017>
70 REM ********************************     <120>
100 PRINT"{CLR,2DOWN,2SPACE}IST 740.914.79
    9 EINE PRIMZAHL ? "                     <203>
110 X=740914799:FOR Y=3 TO SQR(X)STEP 2     <133>
120 Z=X/Y:IF Z=INT(Z)THEN PRINT:PRINT"PRIM
    FAKTOREN ";Y;"*";Z:GOTO 140             <011>
130 NEXT Y:PRINT:PRINT"{2SPACE}"X" IST EIN
    E PRIMZAHL !"                           <142>
140 END                                     <142>
```

Listing 15. »Prim-Test 740...« löst die Frage, die uns von Anfang an interessiert hat

Eine Programmbeschreibung für dieses Programm ist wohl überflüssig.

So, damit wäre dieser Exkurs in die Gefilde der Primzahlen abgeschlossen. Wenn wir unsere erarbeiteten Programme betrachten, so kann man behaupten, daß wir das sehr zeitintensive Problem »Primzahlen« recht gut in den Griff bekommen und alle Möglichkeiten des C 128 beziehungsweise C 64 in Basic ausgeschöpft haben. Es sind nun Maschinen-Sprache Experten gefragt, die auf den in Basic entwickelten Prinzipien schnelle Assembler-Routinen erstellen. Vielleicht fühlt sich ja der eine oder andere unter Ihnen durch diesen Kurs angesprochen und inspiriert.

(Thomas Hansch/kn)

Eingabehinweise

Geben Sie bitte alle Listings bis auf Listing 8 mit dem C 64-Checksummer von Seite 159 ein.
Da Listing 8 ein C 128-Listing ist, sollten Sie für die Eingabe den neuen »Checksummer 128« aus dem C 128-Sonderheft (Sonderheft 22) verwenden. Die Prüfsummen sind nicht identisch mit den Prüfsummen des C 64-Checksummers von Seite 159. Sie können natürlich auf die Prüfsummen verzichten und Listing 8 mit dem normalen Basic-Editor eingeben.

»Maschinen-Power« in Basic

Träumen Sie als Basic-Programmierer auch manchmal von atemberaubender Geschwindigkeit, wie sie die Maschinensprache ermöglicht? Wenn ja, dann kommen Sie im folgenden Artikel voll auf Ihre Kosten. Wir werden dem Basic-Interpreter durch die Anwendung von Betriebssystem-Routinen nämlich ganz schön einheizen.

Der C64 ist in der Grundausstattung nicht gerade mit einem herausragenden Basic bestückt. Will man mehr aus seinem Commodore herausholen, so kann man sich entweder eine der inzwischen zahlreichen Basic-Erweiterungen zulegen, oder sich in die Tiefen der Maschinensprache wagen, um seine eigenen Befehle und Unterprogramme zu erstellen.

Es gibt noch eine dritte Möglichkeit, sozusagen den goldenen Mittelweg, nämlich das Ausnutzen der vorhandenen Interpreter- und Betriebssystem-Routinen von Basic aus. Wenn solche Routinen erst einmal analysiert worden sind, dann wird auch der Basic-Programmierer in die Lage versetzt, diese Unterprogramme ohne Maschinensprache-Kenntnisse zu verwenden.

Kleine »Byte«ologie

Folgende Tatsachen sollten Ihnen jedoch geläufig sein. Wie Sie sicher wissen, speichert Ihr Computer jede Zahl, jeden Buchstaben, jedes Programm, schlicht und einfach alles in Bytes ab. Ein Byte kann bis zu 256 verschiedene Werte annehmen, man kann also Zahlen von 0 bis 255 darin speichern. Um größere Werte verarbeiten zu können, nimmt man einfach zwei Byte zusammen und erhält so die maximale Zahl 65535. Das erste Byte enthält den niederwertigen Anteil (nicht, wie man vielleicht annehmen möchte, den höherwertigen Teil), im folgenden deshalb Low-Byte genannt. Entsprechend heißt das zweite, höherwertige Byte, High-Byte.

Folgende Basic-Zeile wandelt eine Zahl in zwei Byte um:

```
H%=INT(X%/256):L%=X%-H%*256
```

X% ist hierbei die zu wandelnde Zahl; L% und H% das Low- und High-Byte.

Umgekehrt das Errechnen einer Zahl aus zwei Byte:

```
X%=H%*256%+L%
```

Eine Eigenschaft der Maschinensprache ist es, bei Angaben von Speicherbereichen die Endadresse plus 1 anzugeben. Beispiel: Sie wollen den Bildschirm von 1024 bis 2047 speichern. Dann müssen Sie als Startadresse den Wert 1024 und als Endadresse den Wert 2048 (!) verwenden.

Soviel zum nötigen »Grundwissen«. Jetzt geht's los mit dem ersten Tip: Eine häufige Aufgabe, die der Computer zu erfüllen hat, ist das Verschieben von Speicherbereichen. Wollen Sie zum Beispiel einen deutschen Zeichensatz auf dem C64 realisieren, so müssen Sie zuerst den normalen Zeichensatz auf einen freien Platz kopieren und dann dort die gewünschten Zeichen ändern. Eine andere Anwendung ist das Kopieren des Basic-ROMs und des Betriebssystem-ROMs (Kernel) in das darunterliegende RAM, um dort dann Änderungen vorzunehmen (zum Beispiel Eindeutschen des Befehlssatzes). Dabei müssen allerdings über 16000 Byte gelesen und wieder geschrieben werden.

In Basic dauert das eine ganze Weile. Aber der Interpreter stellt uns eine Routine zur Verfügung, die genau dasselbe in Sekundenschnelle für uns erledigt. Hierzu muß man in die Speicherstelle 95 das Low-Byte und in 96 das High-Byte der Anfangsadresse des zu verschiebenden Bereichs eingeben, entsprechend in 90 und 91 die Endadresse. In 88 und 89 schließlich muß die Endadresse (+1) des Zielbereichs angegeben werden. Rufen Sie danach die Routine mit SYS 41920 auf (das Programm in Listing 1 verschiebt einen beliebigen Speicherblock). In den Zeilen 10 und 20 sehen Sie übrigens eine sehr elegante Methode zur Umwandlung einer Adresse in das Low-Byte und High-Byte.

```
1 REM DEMO ZUR CURSORPOSITIONIGRUNG                   <250>
2 REM CREATED BY CHRISTOPH BERGMANN                   <242>
3 REM                                                 <065>
10 POKE 53280,0:POKE 53281,0                          <138>
20 PRINT"{CLR}";                                      <214>
30 FOR X=0 TO 6 STEP.1                                <036>
40 Y=SIN(X)+1                                         <216>
50 POKE 211,X*6:POKE 214,Y*11                         <184>
60 SYS 58732:POKE 783,1:SYS 58634                     <038>
70 PRINT"{WHITE}6{GREY 3}4{GREY 2}E{GREY 1
   }R"                                                <007>
80 NEXT:END                                           <051>
```

Listing 2. Dieses Programm demonstriert die Cursor-Positionierung auf dem Bildschirm

Speicher »herumschieben«

Vom Verschieben von Speicherbereichen zum Verschieben des Bildschirms. Sie können den Bildschirm jederzeit nach oben scrollen ohne die Cursorposition zu verändern, indem Sie einfach die entsprechende Routine mit SYS 59626 aufrufen.

An dieser und den folgenden Routinen sehen Sie übrigens, wie leistungsfähig das Betriebssystem des Commodore 64 zum Beispiel in der Bildschirmsteuerung ist. Man muß es nur zu nutzen wissen. Wenn Sie eine oder mehrere Zeilen auf dem Bildschirm löschen wollen, so können Sie dies ebenfalls dem Betriebssystem überlassen. Einfach die zu löschende Zeile in die Speicherzelle 781 POKEn und die entsprechende Routine mit SYS 59903 starten. Schon ist die gewünschte Zeile vom Bildschirm verschwunden. Wollen Sie mehrere Zeilen löschen, so benutzen Sie eine FOR-NEXT-Schleife. Das sieht folgendermaßen aus:

```
10 FOR T=A TO E : POKE 781,T : SYS 59903 : NEXT T
```

A ist hierbei die Anfangszeile und E entsprechend die Endzeile. Eine weitere, sehr wichtige Unterroutine ist »Cursor Setzen/Holen«. Damit kann man den Cursor auf jede

```
1 REM BLOCKVERSCHIEBUNG                               <198>
2 REM CREATED BY CHRISTOPH BERGMANN                   <242>
3 REM                                                 <065>
10 DEF FN H(X)=INT(X/256)                             <244>
20 DEF FN L(X)=X-FN H(X)*256                          <019>
100 INPUT"ALTE STARTADRESSE";S                        <215>
110 INPUT"ALTE ENDADRESSE+1";E                        <089>
120 INPUT"NEUE ENDADRESSE+1";N                        <212>
200 POKE 95,FN L(S):POKE 96,FN H(S)                   <221>
210 POKE 90,FN L(E):POKE 91,FN H(E)                   <202>
220 POKE 88,FN L(N):POKE 89,FN H(N)                   <136>
300 SYS 41920:END                                     <098>
```

Listing 1. Routine zur Verschiebung von beliebigen Speicherbereichen

beliebige Position des Bildschirms setzen. Einfach die Zeile in 214 und die Spalte in 211 schreiben und... Halt! Machen Sie nicht den Fehler und rufen Sie sofort die Routine auf. Sie sollten zuvor mit POKE 783,1 die Routine auch auf »Setzen« schalten. Dann können Sie mit SYS 58640 das erste Unterprogramm starten (das Listing 2 demonstriert dies anhand einer kleinen Sinuskurve).

Für die zweite Funktion der Unterroutine, nämlich »Cursorposition holen«, müssen Sie lediglich POKE 783,0 und SYS 58640 eingeben, danach steht die Zeile des Cursors in der Speicherzelle 781 und die Spalte in 782. Damit läßt sich zum Beispiel ein Menü, aus dem man mit einem Cursor auswählen kann, sehr gut realisieren.

Sicher ist es Ihnen schon einmal passiert, daß Sie alle möglichen Parameter zur Bildschirmausgabe verändert haben (zum Beispiel Rahmen-, Hintergrundfarbe, Lage des Bildschirms, und so weiter...) und nicht mehr wissen, wie die Anfangswerte ausgesehen haben. Abgesehen davon ist es sehr mühsam, alles »per Hand« wieder zurückzustellen. Durch Aufruf der Routine »Bildschirm-Reset« mit SYS 65409 wird alles wieder in den Ausgangszustand (wie nach dem Einschalten, also dunkelblauer Hintergrund, hellblaue Rahmen- und Zeichenfarbe, und so weiter...) zurückgesetzt. Dies sollten Sie übrigens zu Beginn jedes Programms einmal durchführen, da ja der Benutzer vorher alles verstellt haben könnte.

Wenden wir uns nun ab von der Bildschirmsteuerung hin zu anderen Betriebssystem-Routinen. Zur Fehlersuche zu gebrauchen ist eine Routine, die die aktuelle Zeilennummer, in der sich das Programm gerade befindet, ausgibt. Durch SYS 48578 erfolgt die Ausgabe »in xxxx«, wobei es sich bei »xxxx« um die aktuelle Zeilennummer handelt.

Das laufende Basic-Programm wird danach ganz normal fortgesetzt.

Wenn Sie nicht Fehler verhindern oder aufspüren, sondern produzieren wollen, so hilft Ihnen die folgende Routine weiter. POKEn Sie einen Wert zwischen 1 und 30 in die Speicherzelle 781 und starten Sie das Unterprogramm mit SYS 42039. Schon bricht Ihr Programm mit der der Nummer entsprechenden Fehlerausgabe ab.

Alle Kanäle schließen

Nun wieder zu etwas Nützlichem: Wenn Sie in einem Programm mehrere Ein- oder Ausgabekanäle mit OPEN eröffnet haben, so ist es etwas mühsam, alle Kanäle wieder mit CLOSE zu schließen. Das Betriebssystem hat hierfür ein Unterprogramm, das alle möglichen Kanäle auf einmal schließt. Der Aufruf erfolgt mit SYS 65511.

Wie Sie sicher wissen, kann ein Programm nicht nur mit LOAD "Name",8 (hier wird das Programm immer ab dem Basic-Anfang, der bei 2049 liegt, geschrieben), sondern auch mit LOAD "Name",8,1 geladen werden. Bei letzterem lädt das Betriebssystem ein Programm an die ursprünglich angegebene Stelle im Speicher. Die Startadresse wird

```
1 REM SPEICHERBEREICH ABSPEICHERN                <181>
2 REM CREATED BY CHRISTOPH BERGMANN              <242>
3 REM                                            <065>
10 DEF FN H(X)=INT(X/256)                        <244>
20 DEF FN L(X)=X-FN H(X)*256                     <019>
100 INPUT"STARTADRESSE";S                        <085>
110 INPUT"ENDADRESSE+1";E                        <230>
120 INPUT"NAME{8SPACE}";N$                       <146>
150 SYS(57812)N$,8                               <215>
200 POKE 193,FN L(S):POKE 194,FN H(S)            <206>
210 POKE 174,FN L(E):POKE 175,FN H(E)            <222>
300 SYS 62957:END                                <198>
```

Listing 3. Speichern eines beliebigen Speicherbereichs Kassette oder Diskette

beim SAVEn mitgespeichert. Dies kann man sehr nutzbringend anwenden, denn es muß sich ja nicht unbedingt um ein Maschinenprogramm handeln. Man kann vielmehr jeden beliebigen Speicherbereich speichern, zum Beispiel Sprite-Daten, den Bildschirmspeicher, HiRes-Grafiken, Variablenwerte, etc. Es ergeben sich wirklich viele Anwendungsmöglichkeiten.

Um so einen Bereich zu speichern, müssen Sie folgendes eingeben: Als erstes »SYS(57812)"Name",g« zum Eröffnen des Programmfiles. Wenn Sie für »g« den Wert 8 verwenden, so speichern Sie das Programm auf die Diskette, beim Wert 1 auf Kassette. Danach müssen Sie die Start-

```
1 REM EINGABE - UNTERPROGRAMM                    <250>
2 REM CREATED BY CHRISTOPH BERGMANN              <242>
3 REM                                            <065>
60000 SYS 42336:E$="":Z=512                      <059>
60010 P=PEEK(Z):IF P THEN E$=E$+CHR$(P):Z=
      Z+1:GOTO 60010                             <135>
60020 RETURN                                     <131>
```

Listing 4. Unterprogramm zur Eingabe einer Zeichenkette in eine Stringvariable

adresse des Speicherbereichs in die Speicherstellen 193 und 194 schreiben sowie die Endadresse in 174 und 175. Zum Schluß rufen Sie die Routine zum Speichern mit SYS 62957 auf. Das Programm in Listing 3 speichert einen angegebenen Bereich wie oben beschrieben.

Der Umgang mit Sprites

Im folgenden ein sehr nützliches Beispiel: Wenn Sie in einem Programm Sprites verwenden, so werden Sie die zugehörigen Daten wahrscheinlich in DATA-Zeilen geschrieben haben, diese dann mit einer FOR-NEXT-Schleife wieder auslesen und an einen bestimmten Speicherplatz POKEn. Das dauert bei vielen Sprites nicht nur sehr lange, es verbraucht auch ungefähr vier- bis fünfmal soviel Platz, wie eigentlich nötig wäre. Wenn Sie die Sprite-Daten allerdings vorher auf Diskette speichern und dann direkt in den Speicher laden, so umgehen Sie beide Nachteile. Dies können Sie am besten folgendermaßen bewerkstelligen:

Laden Sie Ihr altes Programm und starten Sie es. Nachdem die Sprite-Daten an die richtige Stelle gePOKEt wurden, können Sie es unterbrechen und löschen. Tippen Sie nun das Programm in Listing 3 ab und starten Sie es. Als Start- und Endadresse geben Sie die entsprechenden Werte für Ihre Sprites ein. Nach Eingabe eines Namens werden die Daten als Programmfile gespeichert. Laden Sie nun ihr altes Programm. Daraus entfernen Sie nun die DATA-Zeilen und die FOR-NEXT-Schleife. Fügen Sie als erste Zeile folgendes ein:

```
1 IF A=0 THEN A=1 : LOAD "Name",8,1
```

Als »Name« verwenden Sie den Namen, den Sie beim Speichern angegeben haben. So, jetzt ist Ihr neues Programm fertig und Sie können es speichern.

Wenn Sie sich schon einmal über den INPUT-Befehl des Commodore-Basic geärgert haben, weil er verschiedene Zeichen (zum Beispiel Doppelpunkt, Komma, führende Leerzeichen und so weiter) einfach nicht übernimmt, dann ist hier die Abhilfe: Rufen Sie mit SYS 42336 die Eingaberoutine des Betriebssystems auf. Diese Routine schreibt alle Zeichen in einer logischen Bildschirmzeile (maximal 80 Zeichen) in den Basic-Eingabepuffer ab Adresse 512. Daraus kann man nun mit einer einfachen Schleife die Eingabe einlesen. Das Ende wird mit einem CHR$(0) gekennzeichnet (das Programm in Listing 4 zeigt ein Beispiel.)

(Christoph Bergman/N. Heusler)



PEEKs und POKEs mit Effekt

Der C64 besitzt viele nützliche Speicherstellen, mit deren Hilfe Sie Ihrem Computer auch die letzten Geheimnisse entlocken können. Aus diesem Grund haben wir die effektvollsten und interessantesten PEEKs und POKEs für Sie gesammelt und thematisch geordnet.

Viele Effekte und Funktionen lassen sich mit den herkömmlichen Basic-Befehlen nicht realisieren. In solchen Fällen muß man mit PEEK und POKE in das Innenleben des C64 eingreifen, um die gewünschten Reaktionen zu erhalten. Dazu gehören etwa die Programmierung von Sound und Grafik aber auch die Einflußnahme auf wichtige Computerfunktionen der Ein-/Ausgabesteuerung sowie die Manipulation der Speicherverwaltung Ihres C64. Die folgende Liste der wirkungsvollsten PEEKs und POKEs soll Ihnen bei dieser Arbeit zur Hand gehen. In sechs Gruppen aufgeteilt bietet sie zu jedem Thema die nötigen Speicheradressen mit ausführlichen Erläuterungen.

Ein-/Ausgabesteuerung allgemein

ASCII-Code der letzten gedrückten Taste (CHR$(0)=keine Taste gedrückt)
PEEK(197)

Bildschirmcode des Zeichens unter dem Cursor:
PEEK(206)

Cursor blinkt schneller
POKE 788,62

Cursorblinken während Programmablauf einschalten
POKE 204,1

Cursorblinken ausschalten
POKE 204,0

Cursor auf angegebene Position setzen (X=Zeile (0 bis 24); Y=Spalte (0 bis 39)).
POKE 211,X:POKE 214,Y
Nach SYS 58640 erscheint der Cursor auf der entsprechenden Position

Fehlermeldungsausgabe sperren
POKE 768,61

Fehlermeldungsausgabe wieder zulassen mit
POKE 768,139

Hintergrundfarbe bestimmen (Farbcode: 0 bis 15)
POKE 53281,Farbcode

Rahmenfarbe festlegen
POKE 53280,Farbcode

INPUT-Befehl: Ausgabe des Fragezeichens unterdrücken (Achtung: Cursor springt nach <RETURN> nicht mehr in die nächste Zeile).
POKE 19,1

Normales Verhalten bei INPUT wiederherstellen
POKE 19,0

Invers-Darstellung einschalten
POKE 199,1

Insert-Modus ausschalten (vor jeder GET-Anweisung POKE 216,0 verhindert in Eingaberoutinen etc., daß die Betätigung einer Cursortaste nach <SHIFT INS/DEL> Grafikzeichen erzeugt).
POKE 216,0

Joystickabfrage Port 1 (1 oben; 2 unten; 4 links; 8 rechts; 16 Feuer).
PEEK(56320)

Joystickabfrage Port 2 (1 oben; 2 unten; 4 links; 8 rechts; 16 Feuer).
PEEK(56321)

Länge des Tastaturpuffers ändern
(X=Länge des Puffers=Maximalanzahl der bei zu schnellen Eingaben zur Weiterverarbeitung gespeicherten Tasten)
POKE 649,X

Listing ohne Zeilennummern
POKE 22,35

Repeatfunktion für Tasten (X=0 Normalzustand; X=64 Repeatfunktion für alle Tasten ausschalten; X=128 Repeatfunktion für alle Tasten einschalten).
POKE 650,X

Abfrage der Sondertasten
PEEK(653)
(1 = SHIFT-Taste gedrückt; 2 = CBM-Taste; 4 = CTRL-Taste). Sind mehrere Tasten gleichzeitig gedrückt, addieren sich die Werte.

Tastaturpuffer löschen nach jeder GET-Anweisung sorgt dafür, daß im Tastaturpuffer gespeicherte und noch nicht verarbeitete Zeichen gelöscht werden – das »Nachlaufen« von Basic-Programmen bei zu schneller Eingabe wird verhindert.
POKE 198,0

Zeichen in Tastaturpuffer schreiben
POKE 631,ASC("A"):POKE 632,ASC("B")...
ASCII-Codes der Tasten <A>, <B>,... in den Tastaturpuffer schreiben, um entsprechende Tastenbetätigung zu simulieren.

Zeichenfarbe bestimmen (X=Farbcode: 0 bis 15)
POKE 646,X

Zeichensatz-Umschaltung sperren
POKE 657,128

Umschaltung wieder zulassen
POKE 657,0

PEEK(647) Farbe unter dem Cursor: Mit den Werten 0 bis 15 werden die jeweiligen Farben gePOKEt: POKE 647,X (X : 0-15)

Repeat-Verzögerung: Alle eingePOKEten Werte verzögern die Repeat-Fuktion. Höherer Wert = größere Verzögerung: POKE652, X

Horizontal-Position des Lichtgriffels:
PEEK(36870)

Vertikal-Position des Lichtgriffels: PEEK(36871)

Softscrolling: Ein horizontales Softscrolling kann folgendermaßen gePOKEt werden.
POKE 53270,0 = scrollen nach rechts
POKE 53270,1 = scrollen nach links

Blinkfrequenz des Cursors: 56325 Die Blinkfrequenz des Cursors kann in diesem Byte bestimmt werden. Höherer Wert = langsamere Blinkfrequenz. Mit POKE 56325,58 wird die normale Blinkfrequenz festgelegt.

Userport-Abfrage. Mit POKE (56576) kann man die Pins PB0–PB7 vom User-Port (auf der Unterseite der Ports, siehe Handbuch) auslesen. Mit POKE in diese Speicherstelle kann man auch Ausgaben über den User-Port laufen lassen.

Datenrichtungsregister für User-Port: 56578. Jedes der Bits gibt die Datenrichtung für die PINS PB0–PB7 des User-Ports an (Bit gesetzt = Ausgang, Bit nicht gesetzt = Eingang).

Der Tastaturpuffer

1. Die Arbeitsweise des Tastaturpuffers zeigt am besten ein kleines Beispiel, wie die folgende »Mini-Textverarbeitung«:

```
100 GET A$
110 FOR I=1 TO 100:NEXT I
120 PRINT A$;
130 GOTO 100
```

Das Demoprogramm wartet auf eine Taste, gibt das zugehörige Zeichen aus und wartet anschließend erneut auf einen Tastendruck. Wegen der Warteschleife in Zeile 110 ist das Programm zu langsam, um Ihren Eingaben zu folgen, wenn Sie sich bemühen, möglichst schnell zu tippen.

Die vom Programm noch nicht verarbeiteten Zeichen speichert der Computer im Tastaturpuffer ab Adresse 631 für die spätere Verarbeitung. In der Speicherzelle 198 wird die Anzahl der noch zu verarbeitenden Zeichen abgelegt, die sich im Tastaturpuffer befinden. Es gehen also keine Zeichen verloren. Wenn der Benutzer schon längst keine Taste mehr betätigt, erscheinen die Zeichen nach und nach auf dem Bildschirm – das Programm »hinkt hinterher«.

Dieses »Nachlaufen« ist bei Basic-Programmen, die mit GET arbeiten (Textverarbeitung, Eingaberoutine), problemlos zu vermeiden, indem vor jeder GET-Anweisung in Speicherzelle 198 (Anzahl der Zeichen im Tastaturpuffer) der Wert 0 gePOKEt wird. Dem C64 wird damit simuliert, daß keine noch zu verarbeitenden Zeichen im Puffer vorhanden sind, und er wartet auf die nächste Tastenbetätigung. Nachteil: Bei zu schnellen Eingaben gehen die nicht mehr sofort zu verarbeitenden Zeichen verloren, da sie nicht mehr im Puffer »gerettet« werden.

2. »Simulierter Direktmodus«. Verschiedene Anweisungen sind nur im Direkt-, nicht im Programmodus möglich. Zum Beispiel können Sie nur im Direktmodus Programmzeilen löschen oder ändern.

Der Direktmodus kann jedoch im Programm simuliert werden. Die gewünschte Anweisung wird Zeichen für Zeichen – die ASCII-Codes der Zeichen! – in den Tastaturpuffer ab 631 gePOKEt. In 198 wird die Zeichenanzahl gePOKEt und das Programm mit END beendet. Nach dem Beenden eines Programms arbeitet der C64 automatisch die Zeichen im Tastaturpuffer ab, also die von Ihnen dort abgelegten Tasten.

```
100 INPUT"FUNKTION (BSP. Y=2*COS(X))";A$
110 PRINT CHR$(147);:REM BILDSCHIRM LOESCHEN
120 PRINT "210 ";A$:REM FUNKTION IN OBERSTE ZEILE
130 PRINT "RUN 200":REM PROGRAMM AB ZEILE 200
    STARTEN
140 POKE 631,19:REM CODE VON 'CURSOR HOME'
150 POKE 632,13:REM RETURN-CODE
160 POKE 633,13:REM RETURN-CODE
170 POKE 198,3:REM SIMULATION VON 3
    TASTENBETAETIGUNGEN
180 END
190
200 INPUT "X-WERT";X
210 Y=SIN(X):REM DIESE ZEILE WIRD GEAENDERT !!!
```

```
220 PRINT Y
230 GET A$:IF A$="" THEN 230:REM AUF TASTE WARTEN
240 GOTO 100:REM NEUE FUNKTION
```

Wenn Sie dieses Demoprogramm eingeben und starten, werden Sie nach einer Funktion gefragt. Geben Sie zum Beispiel ein Y=2*X, und drücken Sie <RETURN>.

Der Bildschirm wird gelöscht und in der obersten Zeile die Zeilennummer 210 und dahinter Ihre Funktionsvorschrift ausgegeben. In der folgenden Zeile wird die Anweisung RUN 200 ausgegeben

200 Y=SIN(X)
RUN 200

In den Tastaturpuffer werden nacheinander die Codes der Tasten <CURSOR HOME>, <RETURN> und nochmal <RETURN> gePOKEt, in den »Zeichenzähler« 198 entsprechend die Zahl 3.

Nach der END-Anweisung bearbeitet der Computer die im Tastaturpuffer abgelegten Zeichen. Das erste Zeichen – CURSOR HOME – setzt den Cursor auf die oberste Bildschirmzeile. Das zweite Zeichen – RETURN – sorgt für die Ausführung der Anweisung, also die Übernahme der neuen Funktion als Zeile 210.

Der zweite RETURN-Code bewirkt die Ausführung der nächsten Anweisung: RUN 200. Das Programm wird ab Zeile 200 neu gestartet. Es fragt Sie nach einem X-Wert, berechnet mit der geänderten Funktionsvorschrift in Zeile 210 den zugehörigen Y-Wert und gibt ihn aus. Wenn Sie anschließend eine beliebige Taste drücken, geht dieses Spiel von vorne los.

Grafik

(VIC=Speicherzelle 53248)

Grafikmodus
Einschalten: POKE VIC+17,PEEK(VIC+17) OR 32
Ausschalten: POKE VIC+17,PEEK(VIC+17) AND 223

Grafikspeicher
Grafikspeicher-Startadresse ab Adresse 8192: POKE VIC+24,PEEK(VIC+24) OR 8

Grafik löschen
Grafikspeicher löschen (Voraussetzung: Beginn ab 8192)
FOR I=8192 TO 8192+8000:POKE I,0:NEXT I

Multicolor-Modus
Einschalten (Voraussetzung: Grafikmodus wurde zuvor eingeschaltet) POKE VIC+22,PEEK(VIC+22) OR 16

---

1. Grafikspeicher schützen. Wie Sie die Hires-Grafik ein-/ausschalten, den Beginn des Grafikspeichers auf Adresse 8192 legen und den Grafikspeicher löschen, finden Sie in der zugehörigen Abteilung unserer Liste.

Anschließend liegt der Grafikspeicher jedoch mitten im eigentlich für das Basicprogramm und die Variablen verwendeten Speicherbereich. Eine Möglichkeit, dieses Programm zu lösen, besteht darin, den für das Basicprogramm verfügbaren Speicherbereich zu begrenzen und dem C64 mitzuteilen, daß er bei Adresse 8191 endet:

POKE 5,255:POKE 56,31

Diese beiden Befehle sollten sich in der ersten Zeile Ihres Grafikprogramms befinden! Ihnen stehen nun zwar nur noch 7 KByte für Ihr Programm zur Verfügung, zum Experimentieren mit der Grafik reicht das jedoch allemal.

2. Punkte setzen/löschen: Die »Grafik-POKEs« übernehmen zwar alle notwendigen Vorbereitungen, der eigentliche Sinn von Grafikprogrammen, das Setzen oder Löschen von Punkten, fehlt jedoch in der Liste. Die folgenden beiden Formeln setzen voraus, daß den Variablen X und Y die Punktkoordinaten zugewiesen wurden.

AD = 320 * INT(Y/8) + (Y AND 7) + 8*INT(X/8)
BN = 7 - (X AND 7)

Nach diesen Berechnungen kann der betreffende Punkt gesetzt oder gelöscht werden.
Setzen: POKE 8192+AD,PEEK(8192+AD) OR 2↑BN
Löschen: POKE 8192+AD,PEEK(8192+AD) AND (255-2↑BN)

Die obigen Berechnungen funktionieren jedoch nur, wenn die Hires-Grafik mit 320 x 200 Punkten eingeschaltet ist. AD ist dabei die Nummer des Bytes ab der Anfangsadresse des Grafikbildschirms und BN ist die Nummer des Bits in dem Byte Anfangsadresse + AD.

Sprites

(VIC=53248, NR=Spritenummer (0 bis 7))
Breite verdoppeln
POKE VIC+29,PEEK(VIC+29) OR 2↑NR

Höhe verdoppeln
POKE VIC+23,PEEK(VIC+23) OR 2↑NR

Kollision feststellen
PEEK(VIC+30):POKE VIC+30,0
(die Bits der beiden kollidierten Sprites sind gesetzt; in VIC+30 muß (!) anschließend der Wert 0 gePOKEd werden, da diese »Kollisionsspeicherzelle« nicht automatisch gelöscht wird!).

Multicolor-Sprite
Einschalten: POKE VIC+28,PEEK(VIC+28) OR 2↑NR

Spritefarben
Definieren: POKE VIC+39+NR,Farbcode

Sprites einschalten
POKE VIC+21,PEEK(VIC+21) OR 2↑NR

Sprites ausschalten
POKE VIC+21,PEEK(VIC+21) AND 255-2↑NR.

X-/Y-Koordinaten
Festlegen durch POKE VIC+2*NR,X:POKE VIC+2*NR+1,Y

Die »Sprite-POKEs« werden verständlicher, wenn Sie ein wenig über den Umgang des C64 mit Sprites erfahren. Bis zu acht Sprites können gleichzeitig verwaltet werden. Jedes Sprite bekommt eine Nummer zwischen 0 und 7. Für eine Sprite-Funktion ist meist ein Register des VIC-Chips zuständig, zum Beispiel Register 29 für die Verdoppelung der Sprite-Höhe. Um ein Register anzusprechen, müssen Sie die Registernummer zur Basisadresse des VIC-Chips addieren:

53248 (Basisadresse) + 29 (Höhen-Register) = 53277

Die Speicherzelle 53277 beeinflußt also die Sprite-Höhe. Um gezielt ein bestimmtes Sprite ansprechen zu können, ist bei fast allen für die Sprites zuständigen Registern je ein Bit einem Sprite zugeordnet.

Bit 0 Sprite Nummer 0
Bit 1 Sprite Nummer 1
...
...
Bit 7 Sprite Nummer 7

Um nun gezielt die Höhe von Sprite Nummer 5 zu verdoppeln, ohne auch alle anderen Sprites zu beeinflussen, muß Bit 5 von Register 53248 gesetzt werden. Gesetzt beziehungsweise gelöscht werden Bits mit Hilfe der logischen Operatoren OR (Setzen) und AND (Löschen). Das folgende Schema zeigt, wie einzelne Bits gezielt manipuliert werden

Bit NR setzen POKE X,PEEK(X) OR 2↑NR
Bit NR löschen POKE X,PEEK(X) AND 255-(2↑NR)

Zur Verdoppelung der Höhe eines Sprites finden Sie in der Tabelle den POKE:

POKE VIC+29,PEEK(VIC+29) OR 2↑NR

NR ist hierbei die Spritenummer (0 bis 7), VIC die Basis-

adresse des VIC-Chips (53248). Um die Höhe von Sprite 3 zu verdoppeln, setzen Sie einfach das entsprechende Bit Nummer 3 mit der Anweisung
POKE 53248+29,PEEK(53248+29) OR 2↑3

Die Verdoppelung können Sie jederzeit rückgängig machen, indem Sie dieses Bit wieder löschen
POKE 53248+29,PEEK(53248+29) AND 255–2↑3

Dieser kleine »Ausflug« sollte verdeutlichen, wie die Inhalte der Sprite-Register vom C 64 interpretiert werden. Zum Umgang mit Sprites halten Sie sich bitte einfach an die POKE-Liste. Die Anwendung dürfte problemlos sein, nachdem Ihnen nun die Bedeutung der Variablen VIC und NR bekannt ist.

Sound

(SID=54272; alle Angaben beziehen sich auf Stimme 1; für Stimme 2 zur angegebenen Adresse den Wert 7 addieren, für Stimme 3 den Wert 14).

Gesamtlautstärke:
Für alle Stimmen POKE SID+24,X (X zwischen 0 und 15).

Hüllkurve:
DECAY- und ATTACK-Wert festlegen POKE SID+5,X
SUSTAIN- und RELEASE-Wert festlegen POKE SID+6,X

Puls-Pause-Verhältnis
POKE SID+2,LB:POKE SID+3,HB (LB/HB Low- und High-Byte der Pulsbreite).

Schwingungsform
POKE SID+4,PEEK(SID+4) OR 2↑I
(I=4→Dreieck; I=5→Sägezahn; I=6→Rechteck; I=7→Rauschen).

Tonhöhe:
POKE SID,HB:POKE SID+1,LB
HB/LB High- und Low-Byte der gewünschten Tonhöhe (siehe Tabelle im Handbuch).

Programmschutz

Ändern von Programmzeilen verhindern
POKE 813,2

Basic-Programm zerstören
POKE 776,1

LIST sperren
POKE 775,200

LIST wieder zulassen
POKE 775,167

Rechner nimmt keine Befehle mehr an
POKE 120,2

RESET nach Programmende
POKE 768,143

RESET bei LIST-Anweisung
POKE 774,226:POKE 775,252

RESET bei Drücken der RESTORE-Taste
POKE 792,226:POKE 793,252

RESET bei SAVE-Anweisung
POKE 818,226:POKE 818,165

Speichern verhindern
POKE 801,0:POKE 802,0:POKE 818,165
oder POKE 818,116 : POKE 819,196 oder
POKE 818,34 : POKE 819,253

Sperren von Tastatureingaben
POKE 649,0

Wiederzulassen von Tastatureingaben
POKE 649,10

STOP-Taste ausschalten
POKE 788,52 oder POKE 808,225

STOP-Taste wieder einschalten
POKE 788,49 oder POKE 808,237

STOP- und RESTORE-Taste ausschalten
POKE 792,193

STOP- und RESTORE-Taste wieder einschalten
POKE 792,71

Absturz:
Durch EinPOKEn eines beliebigen Wertes in der Speicherstelle 770 erfolgt die Ready-Ausgabe unendlich oft, nur noch Ausschalten hilft

Nach POKE 777,1 wird kein Befehl mehr ausgeführt. Der Cursor befindet sich in der linken Ecke.

Speicherverwaltung

Speicherstelle 1
Inhalt 55 = normal
Inhalt 54 = Basic ausgeschaltet (auf RAM umgestellt)
Inhalt 53 = Basic und Kernel auf RAM umgestellt.
(Es empfiehlt sich dabei, das Basic und das Kernal vorher ins RAM zu POKEn, damit der Computer bei der Umschaltung nicht aussteigt.)

Speicherstelle 43/44
Der Anfang des zur Zeit im Speicher befindlichen Basic-Programms errechnet sich durch
PEEK (43) + PEEK (44) * 256

Speicherstelle 45/46
Das Ende des Basic-Programms erhält man durch
PEEK (45) + PEEK (46) * 256

Speicherstelle 55/56
Mit PEEK (55) + 256 * POKE (56) kann das Ende des Basic-RAMs abgerufen werden.

Speicherstelle 57/58
Die Zeilennummer, bei der nach einer Programmunterbrechung gestoppt wurde, errechnet sich durch
PEEK (57) + 256 * PEEK (58).

Speicherstelle 61/62
Zeiger auf Basic-Statement für CONT: Durch PEEK (61) + PEEK (62) * 256 erhält man die Speicherstelle, die nach dem zuletzt ausgeführten Basic-Befehl liegt, das heißt die Speicherstelle, von der sich der Basic-Interpreter bei CONT den nächsten Befehl holt.
Tip: Bei CONT kommt öfter CAN'T CONTINUE ERROR vor, wenn man nach dem Stoppen ein CLR eingegeben oder in irgendeiner Programmzeile etwas geändert hat. Liest man die Werte mit PEEK (61) und PEEK (62) nach der Unterbrechung aus, dann macht ein CLR oder ähnliches nichts aus, wenn man vor CONT die zuvor ausgelesenen Werte wieder in die Speicherstellen POKEt.

Speicherstelle 63/64
Nummer der aktuellen DATA-Zeile:
Mit PEEK (63) + PEEK (64) * 256 erhält man die Nummer der Data-Zeile, aus der gerade das letzte Datum geholt wurde. (Gut zum Finden von Fehlern in DATA-Zeilen geeignet.)

Speicherstelle 69/70
Zuletzt zugewiesene Variable:
Bei normalen Fließkommavariablen liest man den Wert mit PRINT CHR$(PEEK (69))+CHR$(PEEK (70)) aus.

Bei Integervariablen (zum Beispiel XY%) erhält man den Namen durch PRINT CHR$(PEEK (69)-128)+CHR$ (PEEK (70)-128).
Strings (zum Beispiel VX$) erhält man durch PRINT CHR$ (PEEK(69))+CHR$(PEEK (70)-128).

Speicherstelle 641-644
Start- und Endadresse des Basic-RAMs: Durch Ändern dieser Werte kann man die Größe des Basic-RAMs verändern, zum Beispiel:
POKE 643,0 : POKE 644,128 : SYS 64768 setzt das Ende des Basic-RAMs um 8 KByte nach unten. Anderes Beispiel: POKE 641,0 : POKE 642,16 : SYS 64764 setzt das Basic-RAM um 2 KByte nach oben.

Speicherstelle 781/782
Startadresse ab der ein Programm geladen wird: Durch entsprechende POKE-Werte kann ein Basic-Programm in einen anderen Speicherbereich geladen werden.

Speicherstelle 784/785
USR-Vektor: Erfolgt der Einsprung in ein Maschinenprogramm über den USR-Befehl, so muß die Einsprungadresse vorher in diese beiden Byte gePOKEt werden.

Speicherstelle 788/789
IRQ, Hardware-Interrupt: Das Betriebssystem springt ständig in diese Routine, durch Ändern des Inhalts kann man eigene, »interruptgesteuerte« Maschinenroutinen ständig laufen lassen. POKE 788,49 hebt die Wirkung der STOP-Taste auf. POKE 788,52 schaltet sie wieder ein.

Speicherstelle 792/793
Restore-Vektor: PEEK (792) + PEEK (793) * 256 ergibt die Speicherstelle, an die bei Restore-Tastendruck gesprungen wird.
Beispiel: POKE 792,226 : POKE 793,252 wird beim Drücken der Restore-Taste ein Reset ausgelöst.

Damit beenden wir den PEEK- und POKE-Reigen. Doch lassen sich durch Probieren sicherlich weitere interessante Effekte an Ihrem C 64 realisieren.

(Said Baloui/M. Kohlen/M. Thomas)

Basic-Start-Generator

Maschinenroutinen an beliebigen Startadressen laden wie Basic-Programme und mit RUN starten, bleibt nicht länger ein Wunschtraum. Dieses Programm macht es möglich.

Wie war noch die Startadresse für den Monitor? Sie haben bestimmt auch schon öfter die Einsprungadresse eines Maschinenprogramms vergessen, wenn Sie viel mit Assembler-Programmen arbeiten. Abhilfe schafft diese kleine Routine, die Sie mit dem MSE eingeben müssen.

Das Programm (Listing 1) macht aus einem im Speicher befindlichen Maschinenprogramm ein Programm, das einen normalen Basic-Start besitzt. Es braucht lediglich noch auf den Datenträger gespeichert zu werden — fertig! Das SAVEn geschieht mit SAVE"name",8 (1 für Cassette). Damit entfällt die leidige Frage nach dem SYS-Befehl, die bei einer großen Programmsammlung schon lästig sein kann. Das so präparierte Programm kann normal (mit »,8« oder »,1«) geladen und mit RUN gestartet werden.

```
0990  a9 2c      LDA #2C    > Blockanfang ($082C)
0992  a0 08      LDY #08
0994  85 5f      STA 5F     > Übergabeadresse
0996  84 60      STY 60
0998  a9 ??      LDA #??    > Blockende+1
                              (wird gesetzt)
099A  a0 ??      LDY #??
099C  85 5a      STA 5A     > Übergabeadresse
099E  84 5b      STY 5B
09A0  a9 ??      LDA #??    > neues Blockende+1
                              (wird gesetzt)
09A2  a0 ??      LDY #??
09A4  85 58      STA 58     > Übergabeadresse
09A6  84 59      STY 59
09A8  20 bf a3   JSR A3BF   > Sprung in Verschiebe-
                              routine
09AA  4C ?? ??   JMP ????   > Einsprungadresse
                              (wird gesetzt)
```

Assemblerlisting des Programmkopfes

Ausgenommen sind Programme, die im Bereich von $0800 bis $09af und unter dem Basic- und Kernel-ROM liegen, da hier der Basic-Start-Generator und von ihm verwendete Routinen liegen.

Tips zur Bedienung
1. Laden des zu bearbeitenden Maschinenprogramms
2. NEW eingeben
3. Laden des Basic-Start-Generators mit LOAD"START-GENERATOR",8
4. Starten mit RUN
5. Eingabe von SYS 2089,[anfang],[ende+1],[einsprung]

[anfang]: Beginn des Maschinenprogramms im Speicher
[ende]: Ende des Maschinenprogramms im Speicher
[einsprung]: Einsprungadresse, also der SYS-Wert, mit dem normalerweise das Programm gestartet wurde.

Alle Parameter müssen in dezimaler Schreibweise eingegeben werden, wobei der korrekte Bereich (0 bis 65535) selbst überprüft werden muß. Es wird ein »ILLEGAL QUANTITY ERROR« ausgegeben, falls [anfang] > = [ende] ist.
6. Speichern des erzeugten Programms mit SAVE "name",8

Funktion
Nach der Eingabe und Umrechnung der Parameter in das 16-Bit-Format wird das Quellprogramm direkt hinter den Start-Generator kopiert (ab $08af). Dem Quellprogramm ist nun sein späterer Basic-Programmkopf vorgelagert. Des weiteren wird nun ein Teil des Start-Generators nach $C000 kopiert, der die weitere Arbeit übernimmt.

Jetzt werden der Programmkopf und das dahinter liegende Quellprogramm nach $0800, also dem Basic-Start, kopiert. Anschließend werden die Basic-Pointer (Anfang/Ende des Basic-Programms) und im Programmkopf die Adressen gesetzt. Nach der Meldung »READY TO SAVE« kann das bearbeitete Programm gespeichert werden.

Nun eine nähere Erläuterung zum Programmkopf: Im Basic-ROM gibt es eine Unterroutine, die Speicherinhalte verschieben kann und deren Einsprungpunkt bei $A3BF liegt.

Diese Verschieberoutine macht sich der Basic-Start-Generator im Programmkopf zunutze, so daß das Quellprogramm wieder in seinen Ursprungsbereich kopiert werden kann. Die Übergabe der Kopierparameter erfolgt in den

Adressen:
$5F/$60 — alter Blockanfang
$5A/$5B — altes Blockende + 1
$58/$59 — neues Blockende + 1

Nach dem Start des mit einem Basic-Start versehenen Programms wird die Blockverschieberoutine (siehe auch den Artikel »Maschinen-Power von Basic aus« in diesem Sonderheft) aktiviert und das Programm in seinen Bereich kopiert und gestartet.

Soll das Programm nach RUN nur an seinen ursprünglichen Start verschoben werden, müssen Sie als Einsprungadresse eine Adresse wählen, bei der ein RTS (return to subroutine) steht. Beispielsweise 64922, eine Adresse im Betriebssystem des C 64. RTS entspricht einem RETURN in Basic. Es bewirkt, daß das Programm zwar verschoben, aber nicht gestartet wird. Haben Sie zum Beispiel den SMON mit dem Start-Generator behandelt, meldet sich der SMON nach RUN mit der Registeranzeige, wenn Sie als Einsprungadresse 49152 angegeben haben. Haben Sie 64922 angegeben, kommt die READY-Meldung. Mit SYS49152 können Sie dann den SMON starten.

(Hermann-Josef Rottkemper/Nikolaus Heusler)

```
programm : startgenerator  0801 09b0
-------------------------------------
0801 : 0b 08 0a 00 9e 32 30 36  3c
0809 : 31 00 00 00 a9 37 85 01  a7
0811 : a9 09 8d 20 d0 8d 21 d0  46
0819 : a2 00 bd a2 08 20 d2 ff  4c
0821 : e8 e0 57 d0 f5 4c 74 a4  46
0829 : 20 90 08 86 fa 85 fb 20  70
0831 : 90 08 86 fc 85 fd 20 90  f0
0839 : 08 86 fe 85 ff a5 fb c5  9d
0841 : fd 90 0b d0 06 a5 fa c5  68
0849 : fc 90 03 4c 48 b2 a9 af  f8
0851 : 85 39 a9 09 85 3a a5 fa  b5
0859 : 85 3b a5 fb 85 3c 20 f9  13
0861 : 08 a5 fc 85 2b a5 fd 85  0e
0869 : 2c a5 39 85 fa a5 3a 85  38
0871 : fb a9 00 85 39 a9 c0 85  e1
0879 : 3a a9 f9 85 3b a9 08 85  e3
0881 : 3c a9 83 85 fc a9 09 85  70
0889 : fd 20 f9 08 4c 1f c0 a9  2a
0891 : 00 85 0d 20 fd ae 20 9e  ae
0899 : ad 20 9b bc a5 64 a6 65  b8
08a1 : 60 0e 12 93 9e 20 20 20  ab
08a9 : c2 c1 d3 d4 c1 d2 d4 20  21
08b1 : 20 20 20 2d 20 20 20 31  75
08b9 : 39 38 35 20 c8 2d ca 20  c1
08c1 : d2 4f 54 54 4b 45 4d 50  8f
08c9 : 45 52 20 20 20 92 11 11  40
08d1 : 20 d3 59 4e 54 41 58 3a  20
08d9 : 20 53 59 53 32 30 38 39  5c
08e1 : 2c c1 4e 46 41 4e 47 2c  46
08e9 : c5 4e 44 45 2b 31 2c ca  12
08f1 : 4d 50 2d c9 4e 11 11 99  d0
08f9 : a2 00 a1 3b 81 39 e6 39  5b
0901 : d0 02 e6 3a e6 3b d0 02  63
0909 : e6 3c a5 3b c5 fc d0 e8  37
0911 : a5 3c c5 fd d0 e2 60 a9  fe
0919 : 83 85 3b a9 09 85 3c a5  5c
0921 : fa 85 fc a5 fb 85 fd a9  09
0929 : 00 85 39 a9 08 85 3a 20  45
0931 : 00 c0 a5 39 85 2d a5 3a  ef
0939 : 85 2e a5 2d 8d 16 08 a5  d9
0941 : 2e 8d 18 08 a5 2b 8d 1e  63
0949 : 08 a5 2c 8d 20 08 a5 fe  b7
0951 : 8d 29 08 a5 ff 8d 2a 08  4f
0959 : a2 00 bd 78 c0 20 d2 ff  d2
0961 : e8 e0 12 d0 f5 a9 01 85  14
0969 : 2b a9 08 85 2c 4c 74 a4  5c
0971 : 11 11 12 52 45 41 44 59  fc
0979 : 20 54 4f 20 53 41 56 45  be
0981 : 92 11 00 0b 08 c1 07 9e  e5
0989 : 32 30 36 31 00 00 00 a9  da
0991 : 2c a0 08 85 5f 84 60 a9  af
0999 : ff a0 ff 85 5a 84 5b a9  23
09a1 : ff a0 ff 85 58 84 59 20  f0
09a9 : bf a3 4c ff ff 00 00 20  8d
```

Listing 1. Der Basic-Start-Generator. Beachten Sie bitte die Eingabehinweise auf Seite 159.

Der Griff in die Tips- und Tricks-Kiste

Die Beiträge der Rubrik Tips & Tricks sind oftmals unentbehrliche Helfer bei der Arbeit mit dem C64. Doch nur selten findet man aus der Vielzahl der Tips den richtigen auf Anhieb heraus. Eine Zusammenstellung der besten und interessantesten Tricks schafft hier Abhilfe.

Das lästige Suchen nach den benötigten Tips & Tricks hat ein Ende. Denn in diesem Bericht haben wir die nützlichsten Tips, Tricks und Einzeiler aus der Fülle der 64'er-Magazine gesammelt und thematisch geordnet. Die Palette der Themen reicht von einfachen Helfern für Datasette und Diskette, über Rechenprobleme und Programmierhilfen bis hin zu interessanten Effekten. Des besseren Überblicks wegen finden Sie in Tabelle 1 eine Grobgliederung der Themen mit den entsprechenden Seitenzahlen. Diese Ansammlung der verschiedensten Tips & Tricks sind für den Einsteiger — und nicht nur für diesen — eine reichhaltige Fundgrube und zugleich Anreiz, selbst auf »Entdeckungsreise« zu gehen. Wir wünschen Ihnen viel Spaß beim Ausprobieren und Experimentieren.

(Michael Thomas/sk)

Themengebiet	Seite
Cursorsteuerung	132
Datasette	133
Floppy-Tricks	135
Sound	138
Effekte	139
Scrolling	139
Arithmetik-Tricks	140
Programmiertricks	142
Programmschutz	148

Tabelle 1. Die Themen der einzelnen Tips und Tricks im Überblick

Cursorsteuerung leichtgemacht

Bei professionellen Programmen der PC-Klasse kann der Cursor meist über Eingabegeräte wie die Maus positioniert werden. Daß es auch recht gut mit dem Joystick und dem C 64 funktioniert, beweist dieses Programm.

Haben Sie den kurzen MSE-Lader (Listing 1) eingetippt und gestartet, können Sie den Cursor mit einem Joystick in Port 2 steuern. Das Steuerprogramm befindet sich von Adresse $C000 bis $C066 im Speicher. Die ersten 15 Byte nimmt eine Initialisierungsroutine in Anspruch, die den Interruptvektor auf $C00F legt und das Steuerprogramm in den Kernel-Interrupt einbindet. Die Routine wird mit SYS 49152 aufgerufen. Das Programm benutzt Speicherzelle $02 als Zählregister, da der Joystick nur bei jedem sechsten Interrupt abgefragt wird. In Zeile 300 des Assemblerlistings (Listing 2) wird die Zählvariable um 1 erniedrigt. Ist die Variable 0, wird sie auf 6 gesetzt und in die Steuerungsroutine verzweigt. Ab Zeile 360 wird der Tastaturpuffer auf freien Platz überprüft. Sollte der Puffer voll sein, wird sofort in die Interruptroutine des Betriebssystems ($EA31) gesprungen. In Zeile 400 wird der Joystick abgefragt und das entsprechende Cursorsteuerzeichen in den Akku geladen. Ab 800 wird das Zeichen in den Tastaturpuffer geschrieben und der Pufferzeiger erhöht.

(P. Siepen/sk)

```
 PROGRAMM : CURSOR          C000 C066
-------------------------------------

C000 : A9 0F 8D 14 03 A9 C0 8D  B2
C008 : 15 03 A9 06 85 02 60 C6  41
C010 : 02 F0 03 4C 31 EA A9 06  F2
C018 : 85 02 A6 C6 E0 08 90 03  B7
C020 : 4C 31 EA AD 00 DC 29 01  03
C028 : C9 00 D0 05 A9 91 20 5E  2A
C030 : C0 AD 00 DC 29 02 C9 00  2C
C038 : D0 05 A9 11 20 5E C0 AD  6B
C040 : 00 DC 29 04 C9 00 D0 05  63
C048 : A9 9D 20 5E C0 AD 00 DC  C7
C050 : 29 08 C9 00 D0 05 A9 1D  06
C058 : 20 5E C0 4C 31 EA A6 C6  F4
C060 : 9D 77 02 E6 C6 60 00 A0  C7
```

Listing 1. Listing zu »Cursorsteuerung leichtgemacht«. Das Programm muß mit dem MSE eingegeben werden.

```
10 REM*********************************
20 REM*     CURSOR STEUEREUNG          *
30 REM*                                *
65 REM*        PETER SIEPEN            *
70 REM*                                *
75 REM*      VON-STEPHAN-STR.6         *
80 REM*                                *
82 REM*      4200 OBERHAUSEN 1         *
84 REM*                                *
85 REM*   TELEFON :(0208)/26555        *
90 REM*********************************
95 :
100 SYS9*4096
110 .OPT P,00
111 :
112 :
113 :
114 :
120 *= $C000
125 :
130 :
140 :
150 TEST   = $02          ;ZAEHLVARIABLE
160 JOY    = 56320        ;PORT #2
170 AZITP  = $C6          ;ANZAHL ZEICHEN IM PUFFER
174 :
175 :
180 LDA #<BEGINN;INTERUPTVEKTOR
185 STA $314
190 LDA #>BEGINN;AUF NEUE
195 STA $315      ;ADDRESSE SETZEN
200 LDA #$06      ;ZAEHLVARIABLE
205 STA TEST      ;HOCHSETZEN
210 RTS           ;ZURUECK ZU BASIC
215 :
220 :
225 :
300 BEGINN     DEC TEST
320 :          BEQ START     ;GENUG LEER IRR
330 :          JMP $EA31     ;NEIN WEITER MIT IRR
340 START      LDA #$06      ;ZAEHLVARIABLE HOCHSETZEN
350 :          STA TEST
360 :          LDX AZITP     ;TASTATURPUFFER
370 :          CPX #$08      ;VOLL
380 :          BCC WEITER
390 :          JMP $EA31     ;JA WEITER MIT INTERUPT
400 WEITER     LDA JOY       ;WENN JOY NICHT
410 :          AND #1        ;NACH OBEN
420 :          CMP #0        ;WEITER
430 :          BNE NOBEN
440 :          LDA #"□"      ;WENN JA STEUERZEICHEN
450 :          JSR AUSG      ;AUSGEBEN
460 NOBEN      LDA JOY
470 :          AND #2
480 :          CMP #0
490 :          BNE NUNTEN
500 :          LDA #"▒"
510 :          JSR AUSG
520 NUNTEN     LDA JOY
530 :          AND #4
540 :          CMP #0
550 :          BNE NLINKS
560 :          LDA #"▌"
570 :          JSR AUSG
580 NLINKS     LDA JOY
590 :          AND #8
600 :          CMP #0
700 :          BNE NRECHTS
710 :          LDA #"▐"
720 :          JSR AUSG
730 NRECHTS    JMP $EA31
740 :
750 :
760 :
770 :
780 :
790 :
800 AUSG       LDX $C6       ;X-REG FUER
810 :          STA $277,X    ;INDIZIERTE
820 :          INC AZITP     ;ADDRESSIERUNG
830 :          RTS           ;LADEN
840 :                        ;IN TASTPUFFER
850 :                        ;SCHREIBEN
860 :                        ;ANZAHL ZEICHEN
870 :                        ;IM TASTPUFFER
880 :                        ;ERHOEHEN
890 :
900 :
READY.
```

Listing 2. Assemblerprogramm der Cursorsteuerung

Cursor steuern

Das Betriebssystem des C 64 enthält eine Routine, die man benutzen kann, um den Cursor an eine beliebige Stelle zu setzen. Geben Sie doch mal folgendes ein:

`POKE 214, (Zeile) : POKE 211, (Spalte) : SYS 58640 : PRINT "TEXT"`

(Michael Keukert/sk)

Cursor beschleunigt

Für alle diejenigen C 64-Besitzer, denen die Bewegung des Cursors bisher zu langsam war, gibt es einen speziellen POKE.

Mit `POKE 56325,5` wird der Cursor rasend schnell und flitzt bei Betätigung der Cursortasten nur noch so über den Bildschirm. Wer's lieber gemütlicher mag, der sollte es statt dessen einmal mit `POKE 56325,255` probieren.

(Oliver Bausch/sk)

Kassettenpuffer verlegen

Kürzere Maschinenprogramme werden vom Programmierer gerne im Kassettenpuffer abgelegt. Den Floppy-Besitzer stört dies nicht. Er kann Programme laden und das Maschinenprogramm bleibt ihm erhalten. Nicht so beim Benutzer einer Datasette. Lädt er ein Programm von Kassette in den Bereich ab 828 ($033C), so wird es zerstört. Es gibt aber einen Trick, den Kassettenpuffer an einen anderen Speicherbereich zu legen. In den Speicherstellen 178 und 179 steht der Beginn des Kassettenpuffers. Normalerweise beginnt dieser bei 828. Durch Verändern dieses Vektors können wir den Kassettenpuffer beim Laden von Programmen schonen. Um ihn ans Ende des Speicherbereichs zu legen, schreiben Sie: POKE 179, PEEK (56)–2.

(Herbert Kunz/sk)

Datasette richtig justiert

Jetzt ist Schluß mit »LOAD ERROR« beim Abspielen von fremden Kassetten.

Jeder, der eine Datasette besitzt, weiß, daß diese bei einem anderen C 64-Besitzer meist nicht mit derselben Tonkopfeinstellung justiert ist. Bekommt man eine heißerwartete Kassette mit Spielen oder anderen Programmen, womöglich noch im Turbo-Tape-Format, ist die Enttäuschung groß, wenn man feststellt, daß die Tonkopfjustierung von Aufnahme- und Wiedergabegerät abweicht.

```
10 FOR I=36864 TO 36935:READ A:S=S+A:POKE
   I,A:NEXT                                   <Ø54>
2Ø IF S<>6828 THEN PRINT"FEHLER IN DATA'S!
   ":STOP                                     <Ø34>
3Ø PRINT"{CLR,DOWN,5RIGHT}SYNCHRO-TEST FUE
   R DATASETTEN"                              <Ø41>
4Ø PRINT"{3DOWN,5RIGHT}RUECKSPULEN UND 'PL
   AY' ODER"                                  <ØØ6>
5Ø PRINT"{6RIGHT}'PLAY'+'RECORD' DRUECKEN.   <119>
6Ø PRINT"{2DOWN}SIGNAL{SPACE,RVSON}A{RVOFF
   }USWERTEN ODER{SPACE,RVSON}S{RVOFF}CHRE
   IBEN? ";                                   <123>
7Ø POKE 2Ø4,Ø:GET A$:IF A$="A"THEN SYS 368
   96                                         <188>
8Ø IF A$<>"S"THEN 7Ø                          <174>
9Ø SYS 36864                                  <126>
1ØØ DATA 169,Ø,141,17,2Ø8,12Ø,173,18,2Ø8,7
    4,74,74,74,41,8,9,7,133,1,141,24          <2Ø4>
11Ø DATA 212,74,74,74,141,32,2Ø8,76,6,144,
    234,12Ø,169,7,133,1,169,Ø,141,17          <Ø2Ø>
12Ø DATA 2Ø8,141,32,2Ø8,173,13,22Ø,41,16,2
    4Ø,249,173,32,2Ø8,73,1,141,32,2Ø8         <Ø46>
13Ø DATA 1Ø,1Ø,1Ø,141,24,212,76,45,144,Ø,Ø
    ,1                                        <Ø68>
© 64'er
```

Listing 3. »Synchro Justage« erlaubt Ihnen, Ihre Datasette auf fremde Kassetten einzustellen

»SYNCHRO JUSTAGE« (Listing 3) brennt einer Kassette ein 50-Hertz-Synchronsignal auf, das nach der Übergabe der Kassette mit »SYNCHRO JUSTAGE« auf dem Bildschirm ausgewertet wird. Der Empfänger stellt nun mit der Einstellschraube seinen Tonkopf so ein, daß das Flackern des Bildschirmes minimal wird. Je weniger er flackert, desto besser hat man justiert. Wandert der Synchronbalken nach oben, läuft die Datasette bei der Wiedergabe schneller als bei der Aufnahme und umgekehrt. Dann gilt es, zusätzlich die Motorgeschwindigkeit zu überprüfen und optimal einzustellen, bis der schwarze Balken stillsteht.

(Reinhard Abdel-Hamid/sk)

Programme von Datasette nachladen

Mit Hilfe der folgenden POKE-Befehle wird das nächste Programm von der Datasette nachgeladen und automatisch gestartet:

```
POKE 631,131 : POKE 198,1 : END
```

Funktionsweise:

Dem C 64 wird durch den ersten POKE vorgetäuscht, es würde gerade die SHIFT-RUN/STOP-Taste gedrückt. Der Code dieser Tastenkombination (131) wird in den Tastaturpuffer ab Adresse 631 geschrieben. Daraufhin muß dem Computer noch mitgeteilt werden, daß sich in diesem Tastaturpuffer ein noch nicht bearbeiteter Tastencode befindet (POKE 198,1).

(Hans Ippisch/sk)

Bildschirm auf Kassette/Diskette

Das kurze Maschinenprogramm (Listing 4) dient dazu, den Bildschirminhalt des C64 zu speichern und auch wieder einzuladen. Dies kann mit einem Recorder oder Diskettenlaufwerk geschehen. Dabei werden außer dem eigentlichen Bildschirmspeicher (der an eine beliebige Stelle verschoben sein kann und nicht ab Adresse 1024 liegen muß) auch das Color-RAM und die Tabelle der Doppelzeilenkennzeichnungen mitberücksichtigt. Das Speichern und Laden erfolgt über einen Pufferbereich im »versteckten« RAM ab 40960, da die drei verschiedenen Speicherbereiche in einem Stück gespeichert und außerdem die 1000 Farbnibbles zu 500 Byte zusammengeschoben werden. Das erspart beim Arbeiten mit dem Recorder wertvolle Zeit.

Nach dem Eintippen des Basic-Loaders kann durch »RUN 16« geprüft werden, ob es richtig eingegeben wurde. Trotzdem sollte es vor dem Start gespeichert werden, da auch durch Prüfsummen nicht alle Fehler erkannt werden können.

Ausgegeben werden zwei Adressen für das Laden und Speichern. Das Programm kann an jede Stelle des Arbeitsspeichers geladen werden, in der vorliegenden Version lädt es sich an das Ende des Basic-Arbeitsspeichers. Es ist davon auszugehen, daß das Programm direkt nach dem Einschalten ohne andere Erweiterungen geladen und gestartet wird.

Soll ein Bildschirminhalt auf Diskette gespeichert werden, so erfolgt dies durch

```
SYS 40738 "FILENAME", 8
```

Beim Arbeiten mit der Datasette kann (auch beim Laden) die Gerätenummer entfallen; die Angabe einer Sekundäradresse ist nicht erlaubt.

Durch das Speichern bedingte Betriebssystemmeldungen wie »PRESS RECORD & PLAY« sowie das Scrolling des Bildschirms sind unbedeutend und werden nicht berücksichtigt. Alle Bildschirmdaten werden zuvor in einen Zwischenspeicher übertragen.

Das Laden solcherart gespeicherter Bildschirminhalte geschieht mit

```
SYS 40833 "FILENAME", 8
```

```
Ø REM SCREENSAVER 64                              <Ø57>
1 D=55:A=PEEK(56)*256+PEEK(D)-222:GOSUB 4:
  CLR:S=PEEK(56)*256+PEEK(55)                     <216>
2 FOR I=S TO S+221:READ A:POKE I,A:NEXT:A=
  S+173:D=S+1:GOSUB 4:D=S+96:GOSUB 4              <1Ø1>
3 PRINT"{DOWN}SAVE:"S:PRINT"LOAD:"S+95:END       <255>
4 POKE D,A-INT(A/256)*256:POKE D+1,A/256:R
  ETURN                                           <Ø71>
5 DATA 32,,,162,4,177,7,145,9,2ØØ,2Ø8,249,
  23Ø,8,23Ø,1Ø,2Ø2,2Ø8,242,162,24,181             <234>
6 DATA 217,157,231,163,2Ø2,2Ø8,248,162,218
  ,2Ø2,2Ø2,134,8,177,7,41,15,133,11,232           <117>
7 DATA 232,134,8,177,7,1Ø,1Ø,1Ø,1Ø,5,11,14
  5,9,2ØØ,2Ø8,229,23Ø,1Ø,224,219,2Ø8,224          <23Ø>
8 DATA 169,16Ø,133,8,165,1,72,41,254,133,1
  ,169,7,162,244,16Ø,165,32,216,255,17Ø           <Ø38>
9 DATA 1Ø4,133,1,138,176,1,96,76,249,224,3
  2,,,169,,32,213,255,176,243,165,1,72            <13Ø>
1Ø DATA 41,254,133,1,162,4,16Ø,,177,9,145,
   7,2ØØ,2Ø8,249,23Ø,8,23Ø,1Ø,2Ø2,2Ø8,242         <154>
11 DATA 162,24,189,231,163,149,217,2Ø2,2Ø8
   ,248,162,218,2Ø2,2Ø2,134,8,177,9,145,7         <1ØØ>
12 DATA 232,232,134,8,74,74,74,74,145,7,2Ø
   Ø,2Ø8,235,23Ø,1Ø,224,219,2Ø8,23Ø,1Ø4          <131>
13 DATA 133,1,96,32,87,226,162,1,32,121,,2
   4Ø,3,32,241,183,16Ø,2,32,186,255,173,          <Ø36>
14 DATA 221,73,3,133,8,173,24,2Ø8,41,24Ø,7
   Ø,8,1Ø6,7Ø,8,1Ø6,133,8,16Ø,,132,7,132          <Ø47>
15 DATA 9,169,16Ø,133,1Ø,96                       <Ø51>
16 CLR:FOR Q=.TO 221:READ A:X=A+X:Y=A-Y:NE
   XT                                             <Ø57>
17 PRINT"{DOWN}CHECKSUM "MID$("ERROR OK",1
   -5*(X=27349 AND Y=-217),5):END                 <Ø27>

© 64'er
```

Listing 4. Listing zu »Bildschirm auf Kassette/Diskette«

Zunächst werden die Daten in den Puffer von Adresse 40960 bis 42483 geladen und dann in MSB-Tabelle, Farbspeicher und die momentan gewählte Videomatrix übertragen.

(Ralph Babel/sk)

SAVE – mal etwas anders

Laden Sie das Programm »CLIP« (Listing 5) mit LOAD "CLIP",8,1 und geben Sie danach »NEW« ein. Von nun an können Teile aus Basic-Programmen auf Diskette gespeichert werden. Es gibt vier Varianten, um den Befehl anzuwenden:

```
1. SYS 828,»NAME«,AZ
2. SYS 828,»NAME«,AZ-
3. SYS 828,»NAME«,AZ-EZ
4. SYS 828,»NAME«,-EZ
```

Dabei bedeutet: AZ = Anfangszeile, EZ = Endzeile

Die Parameterverarbeitung entspricht dem List-Befehl. Der Programmteil wird unter »NAME« auf Diskette gespeichert. Dieser Befehl erspart das zeitaufwendige und unkomfortable Löschen von Programmzeilen, um am Ende dieser Prozedur nur die gewünschten Zeilen zu erhalten, die man speichern will.

Zur Funktionsweise wäre lediglich zu sagen, daß aufgrund der angegebenen Zeilennummern die Adressen der Zeilen im Speicher berechnet werden und danach dieser Speicherbereich auf Diskette gespeichert wird.

Die Geräteadresse kann durch POKE 186, (gewünschte Geräteadresse) geändert werden.

(Hermann Schinagl/sk)

```
Name : clip                     Ø33c Ø3cb
-----------------------------------------
Ø33c : 2Ø fd ae 2Ø 57 e2 a9 Ø8  4e
Ø344 : 85 b8 2Ø fd ae fØ 7d 9Ø  77
Ø34c : 4b c9 ab dØ 77 2Ø 73 ØØ  c7
Ø354 : 2Ø c2 Ø3 a5 2b a6 2c 85  ee
Ø35c : c1 86 c2 aØ Ø5 b1 5f fØ  62
Ø364 : Ø8 e6 5f dØ f8 e6 6Ø dØ  bb
Ø36c : f4 a6 6Ø a5 5f 69 Ø7 9Ø  ff
Ø374 : Ø1 e8 85 ae 86 af aØ Ø6  95
Ø37c : a2 ØØ b1 5f 48 8a 91 5f  54
Ø384 : c8 b1 5f 48 8a 91 5f 2Ø  f9
Ø38c : fa f5 aØ Ø7 68 91 5f 88  2b
Ø394 : 68 91 5f 6Ø 2Ø c2 Ø3 a5  18
Ø39c : 5f a6 6Ø 85 c1 86 c2 2Ø  b3
Ø3a4 : 79 ØØ fØ b7 c9 ab dØ 1c  c6
Ø3ac : 2Ø 73 ØØ fØ Ø6 2Ø c2 Ø3  16
Ø3b4 : 4c 5f Ø3 a5 2d a6 2e 85  f1
Ø3bc : ae 86 af 4c fa f5 2Ø 6b  d9
Ø3c4 : a9 4c 13 a6 4c Ø8 af ØØ  f1
```

Listing 5. Mit »CLIP« können Sie einzelne Teile eines Programms speichern

Maschinenprogramme speichern

Mit dem folgenden Einzeiler von Markus Eicher können Maschinenprogramme sehr einfach gespeichert werden, ohne den Basic-Pointer zu verstellen. Die Werte für LE und HE werden berechnet, indem man auf die Endadresse eine 1 addiert.

Le = Low-Byte der Endadresse
He = High-Byte der Endadresse
La = Low-Byte der Anfangsadressse
Ha = High-Byte der Anfangsadresse

Low- und High-Byte einer Adresse werden so berechnet:
Adresse = dez. 2000
Highbyte = INT(2000/256)=78
Lowbyte = 20000-Lowbyte*256=32

Die Variable A$ enthält den Namen der Datei, in die auf der Diskette das Maschinen-Programm abgelegt werden soll. X legt fest, ob das Programm auf Diskette (X=8) oder Kassette (X=1) gespeichert wird.

Falls ein langer Programmname gewählt wird, müssen die Befehle abgekürzt werden.

```
1 sys(57812)a$,x:poke193,ls:poke194,hs:p
oke174,le:poke175,he:sys62957
```

Beispiel 1:

Das Programm auf den Adressen 20000 bis 22000 soll auf der Diskette unter dem Namen »Beispiel1« gespeichert werden. Der Dateiname wird der Variablen $A zugewiesen. Die POKE-Befehle lauten:

POKE193,32:POKE194,78;POKE174,241:POKE175,85

Beispiel 2:

Ein Programm von $C000-$C37E soll auf Kassette gespeichert werden. Die Variable X hat also den Wert 1. Die Low- und High-Bytes sind:

1. von $COOO
   Low = $00 = dez. 0
   High= $C0 = dez. 192
2. von $C37E
   Low = $7E = dez. 126
   High= $C3 = dez. 195

Die POKE-Befehle heißen:

POKE193,0:POKE194,192:POKE174,127:POKE175,195

Nachladen von Maschinenprogrammen

Oftmals möchte man ein Maschinen-Programm, welches beispielsweise bei $C000 steht, laden, während noch ein Basic-Programm im Speicher steht. Lädt man das Maschinenprogramm nun mit

```
LOAD "NAME",8,1
```

werden die Zeiger auf das Programmende des Maschinenprogramms verstellt und es kommt zu einem OUT OF MEMORY ERROR. Nach einem NEW, der dies behebt, ist leider auch das Basic-Programm weg und da der C64 keine OLD-Funktion (zum Zurückholen mit NEW oder Reste gelöschter Basic-Programme) bietet, sieht die Sache leider düster aus.

Folgender Einzeiler behebt die Misere:

```
SYS (57812) "NAME",8,1: POKE 780, 0: SYS (65493)
```

Alle Zeiger verbleiben im ursprünglichen Zustand. Dies Routine sperrt zudem einen Teil der Fehlermeldungen. Es kann beispielsweise kein LOAD ERROR auftreten.

(Michael Patra/sk)

Basic-Befehle als ID

Wer schon immer einmal Basic-Befehle in seinem Directory, besser gesagt an der Stelle, wo man sonst die ID vermutet, sehen möchte, kann dies in einer ganz einfachen Art und Weise erreichen:

Man formatiert eine Diskette ganz normal, jedoch gibt man nicht zwei Ziffern als ID ein, sondern »geSHIFTete« Buchstaben (zwei Stück).

Wenn man danach das Directory lädt und listet, erkennt man als ID zwei Basic-Befehle, die der jeweils angegebenen ID entsprechen. Ein Beispiel:

Man nehme eine leere Diskette und gebe ein: OPEN 1,8,15,"N: Name, <SHIFT Z> <SHIFT X>": CLOSE 1. Dann LOAD"$",8 und LIST. Im Directory stehen als ID jetzt die Befehle »RETURN« und »RESTORE«. Mit diesen und anderen geSHIFTeten Buchstaben und Zeichen lassen sich leicht abwechslungsreiche Directories gestalten.

(Ulrich Heitmann/sk)

File-Einträge mit Sonderzeichen

Mit dem Einzeiler von Ralf Peiler lassen sich Zusätze an Filenamen im Directory anhängen. So wird zum Beispiel aus dem Directory-Eintrag "name" der Directory-Eintrag "name",8: oder "name",8,1. Doch nun der Einzeiler:

```
INPUT"name";A$:OPEN 1,8,15, "R:"+A$+"{SHIFT SPACE}
{CBM D}8{SHIFT @{="+A$:CLOSE 1
```

⟨SHIFT SPACE⟩ teilt dem DOS mit, ein Anführungszeichen zu setzen.
⟨CBM D⟩ ist der Code für »,«. »8« entspricht der normalen »8« und ⟨SHIFT @⟩ ist der Code für »:«. Soll »,8,1« an den Filenamen angehängt werden, so ist ⟨CBM D⟩8⟨SHIFT @⟩« zu ersetzen durch »⟨CBM D⟩8⟨CBM D⟩«. Wie Sie sehen, geht das Anhängen kinderleicht. Verblüffung kann man auch durch ein buntes Directory erzeugen. Allerdings lassen sich nicht sämtliche Farben darstellen. Bisher sind folgende Kombinationen gefunden worden:

```
name {SHIFT SPACE}  {CTRL 2}= name  →  weiß
name {SHIFT SPACE}  {CTRL 3}= name  →  rot
name {SHIFT SPACE}  {CTRL 4}= name  →  blau
name {SHIFT SPACE}  {CTRL 6}= name  →  grün
name {SHIFT SPACE}  {CTRL}  = name  →  clr
name {SHIFT SPACE}  {DEL}   = name  →  del (auch
                                       mehrmals)
name {SHIFT SPACE}  {RVSON} = name  →  bis »PRG«
                                       revers
```

Schreibschutz für Disketten

Ist es Ihnen nicht auch schon einmal passiert: Sie haben in mühevoller Kleinarbeit eine Diskette mit allen wichtigen Programmen zusammengestellt, und wie es der Zufall will, plötzlich befinden sich durch falsche Eingabe Files auf der Diskette, die nicht dort hingehören, oder die Diskette ist neu formatiert. Gut, Files lassen sich löschen. Mit einer neu formatierten Diskette sieht das schon anders aus. Die auf ihr gespeicherten Programme sind verloren.

Der mechanische Schreibschutz (Aufkleber) ist auch nicht der Weisheit letzter Schluß, um solchen Zufällen zu begegnen. Wenn häufiger Daten auf der Diskette geändert werden sollen, muß der Schreibschutz jedesmal entfernt und nach erfolgreicher Änderung wieder angebracht werden. Um diesen mechanischen Schreibschutz zu umgehen, kann eine Diskette auch softwaremäßig geschützt werden (Listing 6).
Wie das funktioniert, soll kurz erklärt werden.

Auf der Diskette ist im 2. Byte auf Spur 18 Sektor 0 das Formatkennzeichen der 1541 gespeichert, und zwar die Zahl $41. Wird in dieses Byte eine andere Zahl geschrieben (mit einem Disketten-Monitor), so funktioniert das Beschreiben einer Diskette nur so lange, bis die Floppy-Station neu initialisiert wird (zum Beispiel aus- und wieder einschalten). Wird jetzt versucht, ein Programm oder Daten zu speichern, meldet die Floppy-Station einen Formatfehler und nichts geht mehr. Dieser Software-Schutz funktioniert auch dann, wenn versucht wird, die Diskette mit
`OPEN 15,8,15,"N:TEST"`
zu formatieren. Das heißt, formatiert wird die Diskette mit dieser Befehlssequenz eigentlich nicht. Es wird nur das Inhaltsverzeichnis und ein Diskettenbereich (BAM) gelöscht, der Informationen darüber enthält, welche Diskettenblöcke belegt sind und welche nicht. Dieser Vorgang nennt sich auch »weich formatieren«.

Soll eine softwaremäßig geschützte Diskette »entschärft« werden, taucht ein Problem auf. Es läßt sich ja absolut nichts mehr speichern beziehungsweise ändern. Man muß also das Betriebssystem der Floppy-Station davon überzeugen, daß das richtige Formatkennzeichen an der richtigen Stelle steht. Dazu existiert im Speicher der Floppy-Station eine Speicherzelle, in die nach jeder Initialisierung das aktuelle Formatkennzeichen gespeichert wird. Es handelt sich um die Speicherzelle $101. Wird in diese Speicherzelle mit dem Befehl
`PRINT#15,"M-W"CHR$(1)CHR$(1)CHR$(1)CHR$(65)`
(Zeile 100) wieder das 1541-Formatkennzeichen ($41 beziehungsweise 65 oder A) geschrieben, läßt sich die Diskette wieder ganz normal beschreiben. Unter anderem kann nun auch wieder das Formatkennzeichen auf der Diskette geändert werden. (Bruno Henze/sk)

```
10 OPEN 15,8,15,"I":OPEN 2,8,2,"#"                <076>
20 PRINT#15,"U1 2 0 18 0"                         <248>
30 PRINT#15,"B-P 2 2":GET#2,FO$                   <113>
50 PRINT"{CLR,2DOWN,3SPACE}ALTES FORMATZEI
   CHEN = ";FO$                                   <189>
55 IF FO$="A"THEN X$="X"                          <031>
56 IF FO$<>"A"THEN X$="A"                         <136>
60 PRINT"{DOWN,3SPACE}NEUES FORMATZEICHEN
   = ";X$;"{LEFT}";                               <146>
70 POKE 198,0:WAIT 198,1                          <042>
80 A$=X$:GET X$:IF X$=CHR$(13)THEN X$=A$:G
   OTO 90                                         <043>
85 PRINT X$                                       <191>
90 IF FO$="A"THEN 200                             <172>
100 PRINT#15,"M-W"CHR$(1)CHR$(1)CHR$(1)CHR
    $(65)                                         <189>
200 PRINT#15,"B-P 2 2":PRINT#2,X$;                <071>
220 PRINT#15,"U2 2 0 18 0":PRINT                  <119>
230 IF X$="A"THEN PRINT"{2DOWN,3SPACE,RVSO
    N}SCHREIBSCHUTZ ENTFERNT"                     <197>
240 IF X$<>"A"THEN PRINT"{2DOWN,3SPACE,RVS
    ON}SCHREIBSCHUTZ AKTIVIERT"                   <229>
250 PRINT#15,"I":CLOSE 2:CLOSE 15                 <047>
```


Listing 6. Schreibschutz für Disketten

Directory ohne Programmverlust

Häufig möchte man sich das Directory einer Diskette ansehen, ohne das gerade im Speicher befindliche Programm zu zerstören. Wenn man das DOS 5.1 nicht geladen hat, behilft man sich meist mit der zeitaufwendigen Zwischenspeicherung des Programms auf der Diskette. Es geht jedoch auch einfacher und schneller. Geben Sie einfach den folgenden Befehl ein:
`POKE 44, PEEK(46) + 1`

Damit wird der Basic-Anfang auf einen freien Speicherbereich gestellt. Sie können jetzt wie gewohnt mit »`LOAD "$",8`«
das Directory laden und anschließend auflisten.

Mit `POKE 44,8` sind Sie dann wieder im eigentlichen Programm.

Gleiches bewerkstelligt auch folgender Einzeiler.

```
0 get#1,a$:a=asc(a$+"☰"):printchr$((a=13
0and13or((31<aanda<95)anda));:goto0
```

Er wird gestartet mit
`OPEN 1,8,2,"$":GOTO 0.`

Der OPEN-Befehl eröffnet eine sequentielle Datei (hier das Directory) zum Lesen. Mit
`GET#1,A$`
wird ein Byte vom Disketten-Puffer geholt und das Kommando `A=ASC(A$+"SHIFT/HOME")`

wandelt ASCII-Zeichen in Zahlen (von 0 bis 255) um. Hat A$ die Länge 0, so behält der Ausdruck trotzdem die Länge 1. Durch

```
PRINT CHR$((A=130 AND 13 OR ((31<A AND A<95) AND A)));
```

wird das zugehörige Zeichen zu A gedruckt, wenn A einen Wert zwischen 32 und 95 hat. Ist A=130, so wird RETURN ausgegeben. Die Steuerzeichen und Grafiksymbole werden durch die Formel innerhalb der Charakterstring-Klammer »herausgefiltert«. Nur Zahlen, Buchstaben und Satzzeichen werden ausgedruckt.

Die Routine kann auch für andere Zwecke angewendet werden.

Es ist ohne weiteres möglich, alle Kommentare, Inhalte von Print-Anweisungen und Texte des auf der Diskette befindlichen Programms auf den Bildschirm zu bringen, da man alle Programme auf der Diskette als sequentielle Datei lesen kann.

Eröffnen Sie die Datei dafür nach dem folgenden Schema:

```
OPEN 1,8,2,"filename"
```

Dann wird das Programm mit
GOTO 0 gestartet. Wenn die Ausgabe beendet ist, wird mit der RUN/STOP-Taste die Endlosschleife wieder verlassen.

(Heinzpeter Oelkers/sk)

Floppy-Lister

Mit einem einfachen SYS-Aufruf können Sie Programme und sequentielle Dateien direkt von Diskette listen. Programme im Speicher bleiben dabei erhalten.

»Floppy-Lister« bietet zwei Möglichkeiten, ein Programm von Diskette zu listen. Basic-Programme und sequentielle Dateien können entweder als Klartext oder als Speicherauszug (Dumps) gelistet werden. Im Dumpmodus werden sämtliche Daten als Hexcodes ausgegeben und, soweit möglich, in ASCII-Zeichen übersetzt. Deshalb eignet er sich besonders zum Analysieren von unbekannten Programmdateien. Maschinenprogramme dürfen grundsätzlich nur auf diese Art gelistet werden, will man einen Absturz vermeiden.

Ist das Ladeprogramm (Listing 7) mit dem MSE eingetippt, kann das Maschinenprogramm auf Diskette oder Kassette gespeichert werden. Der »Floppy-Lister« kann dann direkt geladen (LOAD "name", 8, 1, oder LOAD "name", 1, 1) und mit SYS 49152,"XY:filename" aufgerufen werden. Vergessen Sie nach dem absoluten Laden nicht, den erforderlichen NEW-Befehl im Direktmodus einzugeben, um die Basic-Zeiger in einen vernünftigen Zustand zu bringen.

»X« steht für die Fileart: entweder »P« für Programm oder »S« für sequentielle Datei. »Y« steht für den Modus, in dem gelistet werden soll: »L« für einfaches Listen und »D« für Ausgabe in Form eines Speicherauszugs. Zum Beispiel listet der Befehl SYS 49152,"SL:TESTSEQUENZ" eine sequentielle Datei mit dem Namen »TESTSEQUENZ«. Abkürzungen des Filenamens mit »*« sind erlaubt. Falsche Eingaben werden mit einer Fehlermeldung quittiert.

Mit der CTRL-Taste kann die Ausgabe verlangsamt und mit der RUN/STOP-Taste unterbrochen werden. Eine Fortsetzung erfolgt mit der A-, und vorzeitiger Abbruch des Listens mit der DEL-Taste. Am Ende eines Listings muß immer die SPACE-Taste gedrückt werden.

Das Maschinenprogramm für »Floppy-Lister« liegt im Bereich von $C000 -$C2B2 (49152-49842). An das Programm schließt sich noch ein Pufferbereich an, in dem Daten zwischengespeichert werden. "Floppy-Lister" läuft mit den gängigen Erweiterungen, wie DOS 5.1, Simons Basic und Exbasic.

(B. Schulzki/sk)

Floppy-Zeit verkürzen

Die kleine Routine von Robert Loos dient dazu, die Zugriffszeit der Floppy-Disk 1541 ganz entscheidend zu verkürzen. Eine sichere Funktion der Floppy wird nicht gefährdet, wenn der Schrittmotor, der den Schreib-Lesekopf bewegt, wesentlich schneller arbeitet. Der Schrittmotor wird im Interrupt bedient. Daher genügt es, die Größe des Interruptintervalls zu verändern, um die Drehzahl des Motors zu beeinflussen. Standardmäßig wird etwa alle 15 Millisekunden (hier etwa alle 4 Millisekunden) ein Interrupt ausgelöst, der den Stepper um eine Viertelspur bewegt. Alle Bewegungen des Kopfes werden dadurch fast viermal schneller. Vorteile sind neben der Zeitersparnis:

```
programm : floppylister      c000 c2b5
-----------------------------------

c000 : 20 fd ae 20 9e ad 20 a3  ee
c008 : b6 a5 64 85 a7 a5 65 85  a3
c010 : a8 a0 00 b1 a7 99 1f c3  8a
c018 : c8 c0 03 90 f6 ad 1f c3  f4
c020 : c9 14 b0 1e c9 04 90 24  2a
c028 : a0 00 ad 21 c3 85 a8 ad  be
c030 : 20 c3 85 a7 b1 a7 99 22  8b
c038 : c3 c8 cc 1f c3 d0 f5 4c  a9
c040 : 53 c0 a9 6f a0 c2 20 1e  28
c048 : ab 4c 74 a4 a9 88 a0 c2  b2
c050 : 4c 46 c0 a2 00 bd 22 c3  42
c058 : 8d 1e c3 c9 50 f0 08 c9  5f
c060 : 53 d0 e9 a0 04 d0 02 a0  ba
c068 : 00 e8 bd 22 c3 c9 4c f0  2e
c070 : 06 c9 44 d0 d7 c8 c8 e8  3f
c078 : bd 22 c3 c9 3a d0 cd 8c  eb
c080 : 0e c3 ae 1f c3 a9 2c 48  ca
c088 : 9d 22 c3 e8 ad 1e c3 9d  5a
c090 : 22 c3 68 e8 9d 22 c3 a9  18
c098 : 52 e8 9d 22 c3 a9 00 e8  65
c0a0 : 9d 22 c3 ee 1f c3 ad 28  34
c0a8 : 03 8d 0f c3 ad 29 03 8d  f9
c0b0 : 10 c3 a9 22 8d 28 03 a9  ca
c0b8 : c2 8d 29 03 ad 1f c3 a2  14
c0c0 : 25 a0 c3 20 bd ff a9 02  b1
c0c8 : a2 08 a0 02 20 ba ff 20  ef
c0d0 : c0 ff a2 02 20 c6 ff ac  0a
c0d8 : 0e c3 b9 ab c2 8d 11 c3  10
c0e0 : b9 ac c2 8d 12 c3 6c 11  65
c0e8 : c3 a9 00 8d 13 c3 8d 14  df
c0f0 : c3 8d 15 c3 a2 00 20 0f  00
c0f8 : c2 b0 03 4c e9 c1 20 cf  29
c100 : ff 9d 16 c3 a5 90 d0 05  f8
c108 : e8 e0 08 90 e9 8d 15 c3  5b
c110 : a9 00 e8 e0 08 b0 07 9d  6d
c118 : 16 c3 e8 4c 13 c1 20 4c  2c
c120 : c2 ad 14 c3 20 2a c2 ad  f0
c128 : 13 c3 20 2a c2 a9 3a 20  0d
c130 : d2 ff a2 00 bd 16 c3 20  86
c138 : 23 c2 e8 e0 08 90 f5 20  30
c140 : 4c c2 a9 12 20 d2 ff a2  78
c148 : 00 bd 16 c3 29 7f c9 20  1b
c150 : b0 04 a9 2e d0 03 bd 16  7b
c158 : c3 20 d2 ff e8 e0 08 90  b7
c160 : e8 a9 0d 20 d2 ff ad 15  72
c168 : c3 d0 11 18 ad 13 c3 69  30
c170 : 08 8d 13 c3 90 03 ee 14  81
c178 : c3 4c f4 c0 4c e9 c1 20  12
c180 : 51 c2 20 51 c2 a2 ff 20  e6
c188 : 51 c2 a5 a7 e8 9d b3 c2  68
c190 : e0 03 90 f3 20 51 c2 e8  fe
c198 : a5 a7 9d b3 c2 d0 f5 e8  4b
c1a0 : 9d b3 c2 e8 9d b3 c2 a9  ba
c1a8 : b3 85 5f a0 c2 84 60 20  1c
c1b0 : 37 a5 ad 00 03 48 ad 01  50
c1b8 : 03 48 a9 22 8d 00 03 a9  c6
c1c0 : c2 8d 01 03 20 c3 a6 68  75
c1c8 : 8d 01 03 68 8d 00 03 20  c9
c1d0 : 0f c2 b0 b1 4c e9 c1 20  fe
c1d8 : cf ff 20 d2 ff a5 90 d0  1a
c1e0 : 08 20 0f c2 b0 f1 4c e9  b4
c1e8 : c1 20 cc ff a9 02 20 c3  9f
c1f0 : ff ad 0f c3 8d 28 03 ad  84
c1f8 : 10 c3 8d 29 03 a9 60 a0  b3
c200 : c2 20 1e ab a5 cb c9 3c  28
c208 : d0 fa a9 00 85 c6 60 a5  1b
c210 : cb f0 0e c9 3f d0 08 a5  f6
c218 : cb f0 06 c9 0a d0 f8 38  92
c220 : 60 18 60 48 a9 20 20 d2  6f
c228 : ff 68 48 4a 4a 4a 4a 20  17
c230 : 42 c2 a8 68 29 0f 20 42  1b
c238 : c2 48 98 20 d2 ff 68 4c  b0
c240 : d2 ff 18 69 f6 90 02 69  14
c248 : 06 69 3a 60 a9 20 4c d2  10
c250 : ff 20 cf ff 85 a7 a5 90  a0
c258 : f0 05 68 68 4c e9 c1 60  ce
c260 : 0d 20 20 20 20 12 20 53  43
c268 : 50 41 43 45 20 0d 00 0d  57
c270 : 46 45 48 4c 45 52 3a 20  05
c278 : 4e 41 4d 45 20 5a 55 20  cd
c280 : 4c 41 4e 47 20 21 0d 00  29
c288 : 0d 46 45 48 4c 45 52 3a  bf
c290 : 20 46 41 4c 53 43 48 45  a8
c298 : 53 20 45 49 4e 47 41 42  1f
c2a0 : 45 46 4f 52 4d 41 54 20  97
c2a8 : 21 0d 00 7f c1 e9 c0 d7  5e
c2b0 : c1 e9 c0 32 a9 00 ff 00  77
```

Listing 7. »Floppy-Lister«

Das Laufgeräusch wird angenehm leise und kurz, und im Falle einer Kopfjustage fährt der Kopf mit erheblich verminderter Kraft gegen den Anschlag, so daß die Gefahr einer Dejustage deutlich gemindert ist. (sk)

```
10 OPEN 1,8,15,"M-W"+CHR$(7)+CHR$(28)+CHR$
   (1)+CHR$(15)
20 :
30 REM ZUGRIFFSZEIT DER FLOPPY KUERZER
```

Der VALIDATE-Befehl kann mehr

Datasetten-Besitzer verwenden ihn häufig, Eigentümer einer Floppy-Station hingegen so gut wie nie. Dabei eignet sich der VALIDATE-Befehl auch hervorragend zum Vergleich von Maschinenprogrammen an nahezu jeder beliebigen Stelle im Speicher. Wichtig ist das oft beim Programmieren, wenn man wissen will, ob das Programm auf Diskette tatsächlich mit dem im Speicher stehenden übereinstimmt, oder ob die aktuelle Programmversion schon gespeichert ist.

Der Befehl hierzu lautet ganz einfach:

```
VALIDATE "NAME", 8, 1
```

Sequentielle- und User-Dateien laden

Will man mit dem normalen LOAD-Befehl eine Datei mit dem Zusatz »SEQ« oder »USR« laden, so erhält man die lapidare Meldung »FILE NOT FOUND ERROR«. Dabei ist der LOAD-Befehl ebenfalls leistungsfähiger, als er auf den ersten Blick erscheint.

Schreibt man nach dem Dateinamen — aber noch innerhalb der Anführungszeichen — ein Komma und danach den Dateityp (»S« oder »U«), so wird beispielsweise mit:

```
LOAD "ADRESSEN,S",8
```

eine sequentielle Datei mit dem Namen »Adressen« nach $0801 geladen.

Hilfe bei »file not found«

Manchmal starrt man als stolzer Besitzer einer Floppy-Station auf den Bildschirm und die Fehlermeldung »file not found«. Besonders schlimm ist dies, wenn der Fehler in einem Programm auftrat, man also nicht weiß, welches File denn nun nicht gefunden wurde. Kein Problem: Durch die Eingabe von »SYS 63123« (leicht zu merken) bringt der C 64 die Meldung »SAVING...« und den zuletzt verwendeten File-Namen auf den Bildschirm. Sie können das leicht testen: Tippen Sie »LOAD"XYZ",8« ein (plus RETURN-Taste) und anschließend »SYS 63123«. (sk)

Track-Zerstörer: Kopierschutz

Diese Programmzeile von Jörg Wegmeyer produziert auf dem gewünschten Track der Diskette (Variable T) einen READ ERROR 21. Damit erreicht man einen relativ sicheren Kopierschutz.

```
1 open1,8,15:open2,8,2,"#":print#1,"u1 2
 0";t;0:print#1,"m-e"chr$(163)chr$(253)
```

Geräusche von A bis Z

Durch Veränderung der Filterfrequenz und durch verschiedene Filter lassen sich einfache Geräusche erzeugen.
Zeile 10 S=54272:Rem Basisregister
Zeile 20 FORL=0TO24:POKES+L,0:NEXT

	Zeile 30	Zeile 40	Zeile 50	Zeile 60	Zeile 70	Zeile 80	Zeile 90	Zeile 100
Geräusch:	Frequenz	Hall	Grenzfrequenz	Resonanz	Pass	Wellenform	Schleife und POKE	Warteschleife und GOTO
Schuß	POKE S+0,0: POKE S+1,18	POKE S+5,1*16+11	POKE S+22,110	POKE S+23,15*16+3	POKE S+24,5*16+15	POKE S+4,0: POKE S+4,129	FORJ=1TO 255: POKE S+0,J: NEXT	FORA=1TO 1000: NEXT:GOTO80
Explosion	POKE S+0,0: POKE S+1,6	POKE S+5,2*16+13	POKE S+22,100	POKE S+23,15*16+3	POKE S+24,3*16+15	POKE S+4,0: POKE S+4,129	FORJ=1TO 100: POKE S+0,J: NEXT	FORA=1TO 4000: NEXT:GOTO80
Uhrenschlag	POKE S+0,0: POKE S+1,6	POKE S+5,1*16+10	POKE S+22,110	POKE S+23,15*16+3	POKE S+24,1*16+15	POKE S+4,0: POKE S+4,17	FORJ=1TO 255: POKE S+0,7: NEXT	FORA=1TO 500: NEXT:GOTO80
Brandung	POKE S+0,0: POKE S+1,40	POKE S+5,10*16+12	POKE S+22,0	POKE S+23,0	POKE S+24,0*16+15	POKE S+4,0: POKE S+4,129	FORJ=1TO 255: POKE S+0,J: NEXT	FORA=1TO 3500: NEXT:GOTO80

Um einen Schuß zu erzeugen, muß das Programm zum Beispiel so aussehen:

```
10 S=54272
20 FORL=0TO24:POKE S + L,0: NEXT
30 POKE S + 0,0 : POKE S + 1,18
40 POKE S + 5,1 * 16 + 11
50 POKE S + 22, 110
60 POKE S + 23, 15 * 16 + 3
70 POKE S + 24, 5 * 16 + 15
80 POKE S + 4, 0 : POKE S + 4, 129
90 FORJ = 1 TO 255 : POKE S + 0,J : NEXT
100 FORA = 1 T 1000 : NEXT : GOTO 80
```

(Jürgen Hüsgen/sk)

Buntes Listing

Wenn Sie ein Basic-Listing vor sich haben, werden Sie bemerken, wie schwierig es sein kann, zusammengehörende Unterprogramme zu erkennen. Die Idee ist es nun, diese Unterprogramme einfach mit der gleichen Farbe aufzulisten. Dies können Sie mit Hilfe von »künstlichen Steuerzeichen« (siehe Serie in früheren 64'er-Ausgaben) umständlich oder auch mit List COLOR einfach erreichen.

```
10 DATA 72,201,143,208,11,200,177,95,201,32,240,
   3,141,134,2,136,104,76,26,167
20 FOR I=49152 TO 49171 : READ A : POKE I,A : NEXT
30 POKE 774,0 : POKE 775,192
```

Starten Sie hierzu das kleine Programm. Geben Sie hinter einem REM-Befehl direkt ein Zeichen ein, so wird dieses als Farbcode interpretiert. Wenn kein Zeichen hinter dem REM folgt, wird das folgende Listing schwarz gefärbt. Wenn Sie zwischen REM und dem nächsten Zeichen ein Leerzeichen eingeben, wird in der bisherigen Farbe weitergelistet. (Arno Gölzer/sk)

Die Schüttel-Schrift

Dieses kleine Programm (Listing 8) läßt den Bildschirm in einem vorgewählten Bereich hin- und herschütteln. Es eignet sich auch zum Einbau in eigene Programme. Als erstes müssen Sie Listing 6 abtippen. Danach wird über zwei POKEs festgelegt, welcher Bildschirmbereich »geschüttelt« werden soll:

```
POKE 49317, Wert 1
POKE 49318, Wert 2
```

Die Werte 1 und 2 errechnen sich wie folgt:

Wert 1 = Startzeile (1 bis 25) x 8 + 41

Wert 2 = Endzeile x 8 + 50

Wenn Sie Listing 6 mit »RUN 60000« gestartet und die beiden POKEs errechnet und eingegeben haben, können Sie die eigentliche »Schüttel-Routine« starten: SYS 49152. Wenn es Sie genug geschüttelt hat, beenden Sie den Spuk mit SYS 49299. (Erik Becker/sk)

```
59900 REM                                     <013>
59910 REM DATAS                               <210>
59920 REM                                     <035>
60000 DATA 120,169,40,141,20,3,169,192,141
      ,21                                     <099>
60001 DATA 3,173,165,192,141,18,208,173,17
      ,208                                    <020>
60002 DATA 41,127,141,17,208,169,129,141,2
      6,208                                   <036>
60003 DATA 169,1,133,251,169,8,133,252,88,
      96                                      <054>
60004 DATA 173,25,208,141,25,208,48,7,173,
      13                                      <215>
60005 DATA 220,88,76,49,234,173,18,208,205
      ,166                                    <162>
60006 DATA 192,176,70,165,251,201,1,240,33
      ,165                                    <215>
60007 DATA 252,141,22,208,201,7,240,11,230
      ,252                                    <019>
60008 DATA 173,166,192,141,18,208,76,129,2
      34,169                                  <093>
60009 DATA 1,133,251,173,166,192,141,18,20
      8,76                                    <194>
60010 DATA 129,234,165,252,141,22,208,240,
      11,198                                  <076>
60011 DATA 252,173,166,192,141,18,208,76,1
      29,234                                  <117>
60012 DATA 169,0,133,251,173,166,192,141,1
      8,208                                   <142>
60013 DATA 76,129,234,169,8,141,22,208,173
      ,165                                    <167>
60014 DATA 192,141,18,208,76,129,234,120,1
      69,0                                    <126>
60015 DATA 141,26,208,169,49,141,20,3,169,
      234                                     <106>
60016 DATA 141,21,3,88,96,40,82               <132>
60100 REM                                     <215>
60110 REM EINLESEN DER DATEN                  <187>
60120 REM                                     <235>
61000 FOR I=49152 TO 49318                    <057>
61100 READ A                                  <177>
61200 POKE I,A                                <129>
61300 S=S+A                                   <015>
61400 NEXT                                    <193>
61500 IF S<>22085 THEN PRINT"FEHLER IN DAT
      AS !":END                               <006>
61600 CLR                                     <243>
```


Listing 8. Die »Schüttelschrift«

Explodierender Bildschirm

Dieser Basic-Einzeiler bewirkt, daß der Bildschirm wie bei professionellen Spielen vibriert (zum Beispiel bei Explosionen):

```
0 FOR A=0 TO 15:POKE 53270,A:NEXT:GOTO 0
```

(Jan Melichar/sk)

Bildschirmeffekt

Zwei interessante Bildschirm-Flimmer-Effekte zum Ausprobieren:

```
10 FOR I=18 TO 30:POKE 53265,I:NEXT I:GOTO 10
10 FOR I=1 TO 255:POKE 53270,I:NEXT I:GOTO 10
```

oder beide Effekte kombiniert:

```
10 FOR I=18 TO 30:POKE 53270,I:POKE 53265,I:
NEXT I:GOTO 10
```

(Tzimas Kosta/sk)

Soft-Scrolling

Mit diesem Einzeiler von Georg Brandt kann ein beliebiger Text von rechts nach links punktweise über den Bildschirm geschoben werden. Die Variable A mit dem Wert 53270 bildet das Register für horizontales Smooth-Scrolling. Die Variable L enthält die Anzahl der Zeichen (L=40 für die gesamte Bildschirmbreite, weniger für kleinere Textausschnitte), die gleichzeitig auf dem Bildschirm erscheinen sollen. In der Variablen A$ erwartet die Routine den zu zeigenden Text. Das letzte Zeichen sollte ein Leerzeichen sein. Das rechtsbündige Zeichen des Strings steht zweimal auf dem Bildschirm, da es nicht gelöscht wird.

Das Programm arbeitet nach folgendem Prinzip: Der Text wird auf den Bildschirm ausgegeben. Dann wird der Bildschirminhalt mit Hilfe des Smooth-Scrolling-Registers punktweise nach links gezogen, bis er um sieben Punkte verschoben ist. Nun wird der gesamte Text nach links geschoben und das Scroll-Register zurückgesetzt. Dadurch scheint der Text um den achten Punkt verschoben zu sein.

Wichtig:
- Vor Programmaufruf sollte der Bildschirm gelöscht werden, da sonst auch der restliche Bildschirminhalt verschoben wird.
- Auch andere Steuerzeichen außer dem hier verwendeten HOME können eingesetzt werden, um den Text zu positionieren. (sk)

```
1 FOR R=1 TO LEN(A$):FOR I=207 TO 200 STEP
  -1:PRINT"{HOME}"MID$(A$,R,L):POKE A,I:NE
  XT I,R
10 :
20 REM SCROLL
```

Scrollen in x-Richtung

Bei diesem Einzeiler von Hans-Peter Harmann kann die Geschwindigkeit des Scrollers je nach Anwendung neu eingestellt werden. Die Verzögerungsschleife

FOR G=0 TO 3:NEXT

muß nur entsprechend abgeändert werden. Durch den Wert 3 wird nur eine minimale Verzögerung erreicht. Experimentieren Sie ruhig mal damit, mit welchen Verzögerungswerten der Bildschirm wie schnell gescrollt wird.

```
10 fort=1to7:poke53270,t:forg=0to3:next:
next:onagoto10:fory=1024to2023:pokey,194
:next:a=1:goto10
20 rem
```

Scrollen bei bleibendem Text

Dieses kleine Programm stammt von Peter Eckart. Es bewirkt, daß nur Teile des Bildschirmes gescrollt werden. Die Zahl, die Sie in Speicherzeile 59693 POKEn, legt fest, wie viele Zeilen am oberen Rand stehenbleiben. So kann man die Kopfzeile einer Tabelle auf dem Bildschirm festhalten oder eine Information, die für ein Programm wichtig ist (Tastenbelegung, Erklärung der wichtigsten Befehle oder anderes). Beendet wird dieses Programm durch den Befehl POKE 1,55.

Wenn Sie eine »0« gePOKEt haben, wird eine Zeile auf dem Bildschirm festgehalten. Beim POKEn einer »1« zwei Zeilen, beim POKEn einer »3« ...

Obwohl diese Routine sehr kurz ist, braucht sie sehr lange zur Ausführung.

```
1 fori=40960to49151:pokei,peek(i):next:f
ori=57344to65535:pokei,peek(i):next:poke
59639,10:poke1,53
```

Hexadezimal – Dezimal

Immer mehr Leser fragen uns, wie sie möglichst einfach zwischen den beiden Zahlensystemen Dezimal und Hexadezimal umrechnen können. Wir wollen versuchen, einen leicht verständlichen Lösungsweg zu erarbeiten.

1. Was ist das Hexadezimal-System?

Im normalen Zehnersystem repräsentiert jede Stelle einer Zahl eine Zehnerpotenz. Die Zahl 4714 läßt sich auch als Summe von Zehnerpotenzen schreiben.

$4714 = 4x10^3 + 7x10^2 + 1x10^1 + 4x10^0 = 4x1000 + 7x100 + 1x10 + 4x1$

Beim Hexadezimalsystem wird nun jede Stelle einer Zahl nicht mehr durch eine Zehner-, sondern durch eine Sechzehnerpotenz repräsentiert. Auch hier wieder ein Beispiel: Die Hexadezimalzahl 0324 bedeutet nichts anderes als $3x16^2 + 2x16^1 + 4x16^0$ (= 3x256+2x16 +4x1).

Dies hat aber noch weitere Konsequenzen: Im Dezimalsystem wird eine Stelle immer von 0 bis 9 (insgesamt 10 Ziffern) durchgezählt, bevor die nächste Stelle um 1 erhöht wird. Also

00 01 02 03 04 05 06 07 08 09 10...

Im Hexadezimalsystem jedoch wird eine Stelle um 16 Werte erhöht, bevor zur nächsten Stelle ein Wert hinzugefügt wird. Da aber unsere Ziffern von 0 bis 9 dazu nicht ausreichen, wurden zusätzlich die Buchstaben A bis F herangezogen. Sie vertreten die Zahlenwerte 10 bis 15 (von 0 bis 15 sind es 16 Werte!). Es bedeuten:

A = 10, B = 11, C = 12, D = 13, E = 14, F = 15

Die Dezimalzahl 10 ist also gleichwertig mit dem Hexadezimalwert 0A. Damit wären wir auch schon bei der Umrechnung.

```
— Hex X$ nach dezimal X:10
x=0:fori=ltolen(x$):
x0=asc (mid$(x$,i,l)):x=16*x
+x0-48+(x0>64)*7: next
```

und – Dezimal X nach hex X$:

```
— Dezimal X nach hex X$:10
x$="":fori=lto4:x0=x/16:
x=x-int(x0)*16:x$=chr$
(48+x-(x>9)*7+x$:x=
x0:next
```

2. Dezimal – Hexadezimal

Wenn wir eine Dezimalzahl in Hexadezimal umrechnen wollen, so bauen wir den Hexwert Stelle für Stelle von links nach rechts auf.

Nehmen wir also an, wir möchten die Dezimalzahl 41717 in Hexadezimal umrechnen. Dazu teilen wir sie erst einmal durch 16^3

$41717 : 16^3 = 10,1848145$

Uns interessiert hier nur die Vorkommastelle 10. Sie ist gleichbedeutend mit dem Hexwert **A**. Er bildet die letzte Stelle unserer Hexadezimalzahl.

Nun müssen wir von unserer Dezimalzahl $10x16^3$ abziehen. Also $41717 - 10x16^3 = 757$

Um die nächsten Stellen unserer Hexzahl zu erhalten, führen wir diese Prozedur nun noch mit 16^2 und 16^1 durch:

$757 : 16^2 = 2,9703125$ (=**2**)

$757 - 2x16^2 = 245$

$245 : 16^1 = 15,3125$ (=**F**)

$245 - 15x16^1 =$ **5**

Als endgültige Umrechnung der Zahl 41717 ins Hexadezimalsystem erhalten wir also **A2F5**.

3. Hexadezimal – Dezimal

Diese Umrechnung ist schon wesentlich einfacher. Um die Hexzahl A2F5 wieder zurückzurechnen, geht man wie folgt vor:

$Ax16^3 + 2x16^2 + Fx16^1 + 5x16^0$

Da man aber mit den Buchstaben A und F nicht rechnen kann, müssen diese als Dezimalzahlen angegeben werden.

$10x16^3 + 2x16^2 + 15x16^1 + 5x16^0$

Wenn Sie dies auf Ihrem C 64 einmal ausrechnen, werden Sie als Ergebnis wieder die Zahl 41717 erhalten! (sk)

Umwandlung beliebiger Zahlensysteme

Die folgenden beiden Einzeiler von Martin und Hartmut Sprave dienen zum Umwandeln von Dezimalzahlen in Zahlen beliebiger Basis und umgekehrt. Man kann die beiden Programme kombinieren und erhält so eine Umwandlungsroutine zwischen verschiedenen Zahlensystemen. Beide sind auch als Unterroutine in einem Programm denkbar.

Zur Umwandlung dezimal/beliebig:
Die Variable D enthält eine Dezimalzahl beliebiger Größe. In der Variablen B muß die Basis angegeben werden, die der umgewandelten Zahl zugrundeliegt. Das Ergebnis steht in Z$.

Zum Programm: Die Dummy-Schleife (von 0 bis 0) wird benutzt, um später wieder mitten in die Zeile springen zu können. An jeder Stelle wird die Zahl in der Variablen D durch die Basis B geteilt. Dadurch wird die unterste Ziffer abgeschnitten. Die jeweils niederwertigste Stelle ist der ganzzahlige Rest dieser Division und steht in S. Dieser wird in ASCII-Code umgerechnet. Durch den CHR$-Befehl wird der Code zu einer Zeichenkette. Diese wird vorne an Z$ angehängt. Die letzte höchstwertige Ziffer ist erreicht, wenn D<1, da die nächste Stelle dann 0 ist. Solange D>=1 ist, ist die Endbedingung noch nicht erreicht und die FOR-Schleife wird weiter durchlaufen (bis P+1<=0).

```
10 z$="":forp=0to0:d=d/b:s=(d-int(d))*b:
z$=chr$(55+s+7*(s<10))+z$:p=-d:next
40 rem in 10 dez in beliebig
```

Zur Umwandlung beliebig/dezimal:

Diese Routine wandelt eine beliebig große Zahl, deren Basis in der Variablen B steht, in eine Dezimalzahl um. Die Zahl selbst muß in $Z stehen. D wird mit 0 initialisiert. Die Schleifenvariable S dient als Zeiger auf die einzelnen Stellen von $Z. Diese werden nacheinander in ASCII-Code umgewandelt. Der Code für Null (48) wird subtrahiert und das Ergebnis in H zwischengespeichert. Man multipliziert die umgewandelten Stellen (in D) mit B und erhöht sie dadurch um eine Potenz dieser Basis. Dann addiert man die aktuelle Stelle (H) dazu. Bei Darstellung durch einen Buchstaben (H>9) ist aufgrund des ASCII-Codes noch die Subtraktion einer 7 notwendig. Die Schleife wird so lange durchlaufen, bis die niederwertigste Stelle erreicht ist.

```
20 d=0:fors=1tolen(z$):h=asc(mid$(z$,s))
-48:d=d*b+h+7*(h>9):next
50 rem in 20 beliebig in dez
```

Die Fakultätsfunktion

Dieser Einzeiler von Detlev Marks berechnet Fakultäten besser als ein Taschenrechner (größer als »69«). Der C 64 besitzt standardmäßig keinen Befehl zur Berechnung der Fakultät.

```
10 rem fakultaeten
20 :
30 inputa:frb=1toa:c=c+log(b):next:c=c/l
og(10):print10↑(c-int(c));"e";int(c):run
40 :
```

Die Eingabevariable ist A. B dient als Zählvariable und C als Rechen- und Ausgabe-Variable.

Dividieren mit beliebig vielen Nachkommastellen

Die Variable Z enthält die Zahl, die geteilt wird, die Variable D den Dividenden, durch den geteilt wird. In N kann die Anzahl der gewünschten Nachkommastellen angegeben werden. Die Variable E% enthält das Ergebnis ohne Rest. Es wird ausgegeben. R enthält den Rest und in Q% steht der Quotient (beschränkt auf Vorkommastellen).

Die Vorkommazahl erhält man durch Q=INT(Z/D), hier vereinfacht durch Q%=Z/D. Q% wird ausgegeben. Dann wird der Rest (Z-Q%*D) berechnet. Die Schleife wird n mal durchlaufen (N=Anzahl der Nachkommastellen).

Analog zu Q% wird E%=R*10/D berechnet. E% wird ausgegeben als 1. Stelle. Der neue Rest wird gebildet durch:
R (neu) = 10*R (alt) - D*E%.

Nun kann die Schleife durchlaufen werden, bis alle Nachkommastellen ausgegeben wurden.

Beispiel: Wie lautet die 85. Nachkommastelle von 116/13?
Z=116, D=13, N=85 sind die Variablenwerte. Das Programm liefert das Ergebnis 9.

Der Autor dieses Programms ist Heinz Bauschke.

```
0 inputz,d,n:q%=z/d:printq%:r=z-d*q%:for
i=1ton:e%=r*10/d:print"▮▮";e%;:r=10*r-d*e
%:next
10 rem
```

Rundungsfehler

Wie jedermann weiß, ist die Rechengenauigkeit des C 64 ziemlich gering. Manchmal werden gerundete Ergebnisse angezeigt, die zwar richtig aussehen, deren mit Hilfe der INT-Funktion ermittelter ganzzahliger Teil ganz anders aussieht, nämlich um 1 kleiner ist. Ein Beispiel:

1/50*100=2, aber INT(1/50*100)=1!

Dies kann man verhindern, indem man die Zahl erst mit »STR$« in einen String umwandelt, mit »VAL« daraus wieder eine Zahl macht und dann die ganze Zahl ermittelt, also INT(VAL(STR$(1/50*100)))=2.

Dieser Trick ist vor allem nützlich bei Rechnungen, bei denen die 9/10-Rundung schon zu ungenau ist.

(Wolfgang Müller/sk)

Null vor dem Komma

Um Realzahlen besser und übersichtlicher ausgeben zu können, benützt man diese kleine Routine; sie schreibt vor jedes Ergebnis eine 0 wenn es kleiner als 1 ist (also 0.23 statt .23).

Vor dem PRINT muß man folgende Routine aufrufen:

```
1000 XX$ = STR$(X): IF ABS(X) « =1 THEN RETURN
1010 XX$ = LEFT$ (XX$,1) + "0" + RIGHT$ (XX$,LEN
(XX$)-1)
1020 RETURN
```

Danach steht das Ergebnis in der Variablen XX$.

(Kurt Huwig/U.Beenen)

Zahlen rechtsbündig

```
  10 f$="{6 space}":rem fuellstring
  20 inputx:gosub1000:printx$:end
1000 x=int(x*100+.5)/100
1010 x$=f$+str$(x)
1020 ifx=int(x)thenx$=x$+".00"
1030 ifmid$(x$,len(x$)-1,1)="."thenx$=x$+"0"
1040 x$=right$(x$,9)
1050 return
```

Zu Beginn des Programms wird ein Füllstring definiert. Zur Aufbereitung der Variablen X wird ins Unterprogramm verzweigt. Zeile 1000 rundet X auf die Anzahl der Nachkommastellen (hier 2). Zeile 1010 wandelt X in eine Stringvariable und stellt ihr den Füllstring voran. In den Zeilen 1020 und 1030 werden bei Bedarf die Nachkommastellen auf zwei aufgefüllt. Zeile 1040 schneidet den String der passenden Länge heraus, hier sechs Vor- und zwei Nachkommastellen plus Dezimalpunkt. (H. G. Sander/sk)

Formatierte Ausgabe

Dieser Einzeiler gibt Zahlen unabhängig von Länge und Vorzeichen rechtsbündig aus. Die Tabulatorfunktion von Volker Walter ist sicher beim Aufbau von Tabellen nützlich.

Die Variable A enthält die Zahl, die ausgegeben wird. Die Zahl »22« legt fest, daß die Kommata der ausgegebenen Zahlen in Spalte 24 stehen (2 Stellen werden vom Komma gebraucht). Die Spaltennummer kann im Listing beliebig gesetzt werden (Zahl im Listing = tatsächliche Spaltennummer - 2).

Ist A > =1, so wird der Zehnerlogarithmus von A berechnet und von der Spaltennummer abgezogen, bei der der Dezimalpunkt stehen soll. Da der C 64 nur den Logarithmus zur Basis e berechnen kann, muß das Ergebnis mit der Konstanten .43429448188 multipliziert werden. Dadurch erhält man den Zehnerlogarithmus.
Diese Berechnung erfolgt durch
INT(LOG(B-(B=0))*.43429448188)*(B< =1).
Ist A<1, so wird zur Spaltennummer eine 1 addiert.
Null bildet einen Sonderfall, da die Null noch vor dem Komma stehen muß.
INT(-B)*(B<1).

```
20 b=abs(a):printtab(int(log(b-(b=0))*.4
3429448188)*(b>=1)+int(-b)*(b<1)+22);a
```

Texte formatieren

Oft hat man das Problem, Texte entweder mittezentriert oder rechtsbündig auszugeben. Mit einer der folgenden kleinen Basic-Zeilen ist dies am einfachsten zu realisieren:
Mittezentrierung:
```
FOR I = 1 TO (40-LEN(A$))/2 : PRINT " ";:NEXT:PRINT A$
```
Rechtsbündig:
```
FOR I = 1 TO 40-LEN(A$):PRINT " ";:NEXT:PRINT A$
```
In der Variablen A$ muß jeweils die zu formatierende Textzeile enthalten sein. (Hans-Peter Harmann/sk)

ASCII-Code in Bildschirmcode umwandeln

Wie wandelt man am einfachsten ein Zeichen vom ASCII-Code (PRINT CHR$(...)) in den Bildschirmcode (POKE...) um? Die wirklich einfachste und genialste Lösung besteht darin, das entsprechende Zeichen auf den Bildschirm zu bringen und dann mit PEEK den Code direkt aus dem Bildschirmspeicher auszulesen. Also:
```
10 PRINT"(HOME)"CHR$(ASCII-Code):PRINT PEEK(1024)
```
Umgekehrt geht es natürlich ebenso einfach:
```
10 POKE 1024,Bildschirmcode:PRINT"(HOME)":OPEN1,3:
   INPUT#1,A$:PRINT ASC(A$+CHR$(0)):CLOSE1
```
Und weil wir gerade bei der Bildschirmverwaltung sind: Um in einem Programm die nächste PRINT-Anweisung zu positionieren, gibt es zwei Möglichkeiten:
1. Man verwendet eine Kolonne von Cursor-Steuerzeichen, oder
2. zwei POKEs und einen SYS-Befehl:
```
POKE211,Spalte:POKE214,Spalte:SYS58640
```
(Hans Ippisch/sk)

»NEW« rückgängig machen

Wie oft hat man »NEW« eingetippt, um hinterher festzustellen, daß man das Programm noch gar nicht gespeichert hat. Mit dieser kleinen Routine können Basic-Programme nach einem »NEW« wieder vollständig zurückgeholt werden.

Man geht dabei so vor: Zuerst einmal muß das Listing 9 eingetippt und gestartet werden. Es erzeugt dann auf Diskette das Programm »AUTO-OLD«. Hat man nun einmal aus Versehen »NEW« eingegeben, so legt man die Diskette mit dem Old-Programm in die Floppy und gibt »LOAD"AUTO-OLD",8,1« ein. Das Old-Programm wird nun geladen und automatisch gestartet. Ein eventuell gelöschtes Basic-Programm ist wiederhergestellt.
(Georg Brandt/sk)

```
 0 :REM     AUTO-OLD BY G. BRANDT                <094>
10 OPEN 1,8,1,"AUTO-OLD,P,W":FOR I=0 TO 40
   :READ A:PRINT#1,CHR$(A);:NEXT:CLOSE 1         <059>
20 DATA 221,2,165,44,160,1,145,43,32,51,16
   5,165,34,24,105,2,133,45,165,35,105           <068>
30 DATA 0,133,46,169,131,162,164,141,2,3,1
   42,3,3,108,2,3,139,227,221,2                  <079>
```

© 64'er **Listing 9. Die Auto-OLD-Routine können Sie mit Hilfe des Checksummers (Seite 159) eingeben**

Betrifft: Joystick

Jeder Neuling unter den C 64-Fans wird sich früher oder später fragen, warum Commodore gleich zwei Joystick-Ports einbaute, es aber versäumte, das Basic des C 64 um eine Funktion zu bereichern, diese Joysticks auch abzufragen. Viele andere Computer haben einen »JOY(X)«-Befehl, der die Richtung angibt, in die der Joystick gerade gedrückt wird.

Auf dem C 64 wurden solche Abfragen bisher mit langwierigen IF-THEN-Sequenzen über die PEEK-Funktion realisiert. Es gibt aber einen sehr viel eleganteren Weg: die DEF FN-Funktion.

Wenn man nämlich am Anfang eines Basic-Programms definiert:
```
DEFFNJOY(X)=INT((LOG(255.5-(PEEK(56322-X)OR224)))/
LOG(2)+2)
```
so läßt sich über »PRINT FN JOY(x)« der Joystick abfragen. »x« gibt dabei an, ob man die Position von Joystick-Port 1 oder 2 wissen möchte. Es entsprechen: 1: Nullstellung, 2: oben, 3: unten, 4: links, 5: rechts und 6: Feuerknopf

Über eine »ON FN JOY(x) GOTO ...«-Anweisung ließe sich dann äußerst schnell in die entsprechenden Unterprogramme verzweigen. (Henning Zipf/sk)

DATA Eingabehilfe

Das kleine Programm ist eine Hilfe zum Eintippen von DATAs. In Zeile 1 wird nach der Startzeilennummer gefragt, ab der die DATAs beginnen und in welcher Schrittweite die Zeilennummer erhöht werden soll.

Nachdem das Programm mit RUN gestartet und die beiden Parameter eingegeben wurden, erscheinen 20 Zeilen mit der gewünschten Startzeilennummer und dahinter das Wort »DATA«.

Diese Zeilen können nun editiert werden.

Das Programm läßt sich beliebig verändern. Statt DATA kann man auch PRINT oder ähnliches einsetzen.

```
1 INPUT "STARTZEILENNUMMER";X: INPUT
"SCHRITTWEITE";Y: FOR Z=1 TO 20
2 PRINT X "DATA": X=X+Y: NEXT:PRINT
"{HOME}{CRSR DOWN}": END
```

(Frank Besler/sk)

DATA-Zeilen

Wer beim Überprüfen von DATA-Zeilen keinen Helfer hat, kann den Computer zu Hilfe nehmen. Man braucht nur einen Sprachsynthesizer (beispielsweise »S.A.M.«) laden und auf Sprachausgabe zu schalten (bei »S.A.M.« RECITER laden und »***« »R« eingeben).

Die Einleseschleife muß dann folgendermaßen aussehen:

```
100 READ X : POKE Y,X : Z=Z+X : A$=STR$(X): SAY A$
: NEXT
```

Zu beachten ist, daß der »S.A.M.« die Zahlen Ziffer für Ziffer ausspricht (one, two, one = 121), nur 64 spricht er als sixty-four. (Kurt Huwig/U.Beenen)

Basic-Programme verbinden

So manchen C 64-Besitzer wird es schon geärgert haben, daß sein Computer keinen MERGE-Befehl besitzt. Mit wenig Aufwand ist es aber dennoch möglich, Basic-Programme aneinanderzuhängen:

1. Im Direktmodus »PRINT PEEK(43); PEEK(44)« eingeben und sich die Ergebnisse merken.
2. Das erste Programm normal laden.
3. Erscheint jetzt nach »PRINT PEEK(45)« eine 0 oder eine 1, dann geben Sie »POKE 43, 256 +PEEK(45) 2 : POKE 44, PEEK(46) - 1 : NEW« ein. Im anderen Fall wird »POKE 43, PEEK(45) - 2 : POKE 44,PEEK(46) : NEW« eingegeben.
4. Nun wird das anzuhängende Programm geladen (Achtung! Das anzuhängende Programm muß die höheren Zeilennummern haben).
5. Jetzt POKEn Sie in die Speicherstellen 43 und 44 die zu Anfang gemerkten Werte.

Beide Programme sind nun verbunden. Wichtig bei der ganzen Prozedur ist, daß noch keine Variablen definiert werden, da das MERGEn sonst nicht richtig funktioniert.

Eine ganz kurze Routine, die sich auch innerhalb von Programmen verwenden läßt, verkettet Programme auf ähnliche Weise.

```
10 a=peek(45)+256*peek(46)-2:poke44,a/25
6:poke43,a-peek(44)*256:print"prg laden
& pO43,1:pO44,8
```

Maschinenprogramme auf Tastendruck

Mit einer kleinen Routine kann man ein Maschinenprogramm mit einem Tastendruck aufrufen. Dafür benutzt man ein Zeichen, das normalerweise nicht oder nur selten verwendet wird. Ich habe mich für das @-Zeichen entschieden.

Im Interpreter existiert eine Schleife, die einen Basic-Befehl holt und ausführt.

```
A7E1 JMP (0308) ; zeigt normalerweise auf A7E4
A7E4 JSR 0073   ; nächstes Zeichen aus Basic-
                  Text holen
A7E7 JSR A7ED   ; Statement ausführen
A7EA JMP A7AE   ; zurück zur Interpreterschleife
```

In den Speicherzellen $0308 und $0309 (776 und 777 dez) liegt ein Zeiger, der normalerweise auf den Beginn dieser Schleife zeigt. Verbiegt man nun den Zeiger auf eine eigene Routine, kann man den Basic-Befehl auf das eigene Zeichen überprüfen.

Wird es erkannt, springt man auf den Anfang des gewünschten Unterprogramms. Wurde das Zeichen nicht vorgefunden, macht man in der Schleife normal weiter.

Dieses Verfahren verwende ich bei der Programmierhilfe »Merge« aus Ausgabe 4/84, die normalerweise mit SYS 50000 gestartet werden muß. Es kann aber auch für andere Maschinenprogramme umgeschrieben werden.

»Merge« belegt den Speicherbereich 50000 bis 50264. Die eigene Routine beginnt auf Adresse 49152 (C000 hex). Der Wert in den Adressen $0308 und $0309 muß deshalb auf C000 abgeändert werden. Der Computer durchläuft dann vor jedem Befehl, den er ausführen soll, folgende Schleife:

```
C000 JSR 0073 ; nächstes Zeichen holen
C003 CMP   40 ; Vergleich mit @-Zeichen
C005 BEQ      ; verzweigen wenn erkannt
C007 JSR 0079 ; Flags setzen
C00A JMP A7E7 ; Rücksprung
C00D JSR 0073 ; nächstes Zeichen holen
C010 JSR C350 ; zur eigenen Routine
C013 JMP A7AE ; Rücksprung
```

Nach dem Drücken von @ und RETURN wird nun das Programm ab Adresse 50000 (C350 Hex) ausgeführt. Auf die anderen Befehle hat diese Routine keinen Einfluß. Eine Hürde gibt es noch zu meistern. Die Änderung der Adressen 0308 und 0309 ist auf der Basic-Ebene mit POKE nicht möglich. Dies ist auch verständlich, da POKE auch ein Basic-Befehl ist und durch die Änderung der ersten Adresse der Einsprung verändert wird.

Deshalb muß diese Adreßänderung in Maschinensprache durchgeführt werden.

```
C100 LDA   00 ; Lade Akku mit 00
C102 STA 0308 ; Speichere Akku nach 0308
C105 LDA   C0 ; Lade Akku mit C0
C107 STA 0309 ; Speichere Akku nach 0309
C10A RTS      ; Rückkehr nach Basic
```

Basic-Lader für Befehlserweiterung

```
240    FOR I= 49152 TO 49152 + 21
250    READ Q : POKE I, Q
260    NEXT
300    FOR I = 49408 TO 49408 + 10
310    READ Q : POKE I, Q
320    NEXT : SYS 49408
11000 DATA 32,115,0,201,64,240,6,32,121
12000 DATA 0,76,231,167,32,115,0,32,80
13000 DATA 195,76,174,167
14000 DATA 169,0,141,8,3,169,192,141,9,3,96
```

Diesen Basic-Lader tippt man hinter das Programm »Merge« und speichert es gemeinsam ab.

Die Zeilennummern sind so gewählt, daß man sie direkt zum Basic-Lader von »Merge« dazutippen kann. In Zeile 10260 müssen aber dann die letzten fünf Nullen gelöscht werden.

Mit SYS 49408 wird die Befehlserweiterung aktiviert und steht dann zur Benutzung bereit. (Patrik Fleig/sk)

Der C 64 als PET

Wenn Sie CBM 2000, 3000 oder 4000 geschriebene Programme auf Ihrem C 64 laufen lassen wollen, müssen Sie umständlich PEEKs und POKEs ändern. Der »Pet-Simulator« nimmt Ihnen diese Arbeit ab.

Ist das Programm (Listing 10) eingegeben und gestartet, werden als erstes die DATAs für das Maschinenprogramm in den Bereich ab Adresse 49152 gePOKEt (SU = Prüfsumme für die Daten). Danach fragt das Programm nach der Zeichenfarbe. Sie werden aufgefordert, eine Zahl zwischen 0 und 15 einzugeben.
(0 = schwarz, 1 = weiß,..., 15 = grau 3).

Bei anschließendem Starten des Maschinenprogrammes wird das Basic-ROM in das darunterliegende RAM gePOKEt (Basic-Interpreter kopieren). Anschließend wird das Bildschirm-RAM von Adresse 1024 nach Adresse 32768 verlegt. Basic-Speicheranfang und -ende werden dem des PET angepaßt. Weiterhin wird in der POKE-Routine des Basic-Interpreters ein Eingriff vorgenommen, nach der der Computer aus dem Interpreter in eine Routine des Maschinenprogramms springt. Hier wird überprüft, ob das Bildschirm-RAM angesprochen wurde. Trifft dies zu, wird die dazugehörige Farb-RAM-Adresse berechnet und der vorher festgelegte Farbwert (Zeichenfarbe) hineingePOKEt. Um das Zurücksetzen des Bildschirmes auf das C 64-Format zu vermeiden (durch Drücken der RUN/STOP- und RESTORE-Tasten), wird die RESTORE-Taste durch Verändern des NMI-Vektors ausgeschaltet. Programme können aber noch mit der RUN/STOP-Taste unterbrochen werden.

Nach Ablauf des Maschinenprogrammes meldet sich der Computer mit »PET-SIMULATOR AKTIV«. Sie können jetzt immer noch die Zeichenfarbe mit POKE 49239, ZF (ZF = Zeichenfarbe – siehe oben) ändern. Wenn Sie jetzt zum Beispiel POKE 32768,1 eingeben, erscheint ein »A« am linken oberen Bildschirmrand in der gewählten Zeichenfarbe. Schlußbemerkung: Bevor Sie das Programm starten, empfiehlt es sich, es vorher zu speichern, da sich das Programm selbständig löscht. (Wolfgang Hopf/sk)

```
1 REM ***********************************     <132>
2 REM *        PET - SIMULATOR          *     <251>
3 REM *                                 *     <052>
4 REM *      (C) BY W. HOPF 1984        *     <218>
5 REM ***********************************     <136>
6 :                                           <238>
7 REM PROGRAMM VOR DEM START ABSPEICHERN      <071>
8 :                                           <240>
10 FOR I=49152 TO 49152+91:READ A:SU=SU+A    <231>
20 POKE I,A:NEXT                              <117>
30 IF SU<>12552 THEN END                      <064>
40 PRINT"{CLR,2DOWN}BITTE WAEHLEN SIE DIE
   ZEICHENFARBE"                              <063>
50 PRINT"{DOWN}(0-15 EINGEBEN)!{DOWN}"        <124>
60 INPUT ZF:IF ZF>=0 AND ZF<=15 THEN POKE
   49239,ZF:POKE 646,ZF:GOTO 80               <089>
70 PRINT"{DOWN}NICHT ERLAUBT":FOR I=1 TO 1
   000:NEXT:GOTO 40                           <066>
80 SYS 49152:POKE 1,54:PRINT"{CLR}PET-SIMU
   LATOR AKTIV":NEW                           <208>
10000 DATA 160,0,132,254,169,160,133,255      <121>
10005 DATA 177,254,145,254,200,208,249,230    <103>
10010 DATA 255,166,255,224,192,208,241        <189>
10015 DATA 169,5,141,0,221,141,24,208,169     <221>
10020 DATA 128,141,136,2,133,56,169,4         <229>
10025 DATA 133,44,169,0,141,0,4,169           <241>
10030 DATA 63,141,37,184,169,192,141,38       <018>
10035 DATA 184,169,193,141,24,3,96,32         <211>
10040 DATA 235,183,24,165,21,201,128,144      <095>
10045 DATA 18,201,132,176,14,24,105,88        <177>
10050 DATA 133,255,165,20,133,254,169,0       <209>
10055 DATA 234,145,254,96                     <037>
60000 :                                       <029>
60010 REM LISTE DER VERWENDETEN COMMODORE-
      STEUERZEICHEN                           <140>
60020 REM "{CLR}" = CLR                       <245>
60030 REM "{DOWN}" = CRSR-DOWN                <186>
```

Listing 10. Listing zu »Der C64 als PET«

Pseudo-Interrupt

Diese Befehlserweiterung erlaubt es, ein Basic-Programm zu jedem beliebigen Zeitpunkt per Tastendruck durch die F1-Taste unterbrechen zu lassen. Es kann dann in eine vorher definierte Basic-Routine gesprungen werden. Diese könnte zum Beispiel den noch freien Speicherplatz oder die Uhrzeit anzeigen.

Das Programm (Listing 11) bitte mit dem MSE eingeben. Es liegt dann im Speicherbereich von 40499 bis 40768. Geladen wird es absolut mit LOAD "PSEUDO-IRQ",8,1. Da das Programm im Bereich für die Basic-Variablen steht, muß es durch POKE 56,158:CLR vor Überschreiben geschützt werden. Nach dem Start durch »SYS 40541« stehen die neuen Befehle zur Verfügung:

!F1JUMP <Zeilennummer> legt fest, in welche Zeile im Falle einer Unterbrechung durch die F1-Taste gesprungen werden soll. Tritt der Befehl mehrmals auf, so gilt die zuletzt angegebene Zeilennummer.

!JBACK bewirkt die Fortsetzung des Basic-Programms ab der Stelle, an der unterbrochen wurde.

!SF1 verhindert Unterbrechungen. Dies kann zum Beispiel beim Aufbau einer Grafik oder bei Arbeiten mit der Diskettenstation wichtig sein. !CF1 läßt gesperrte Unterbrechungen wieder zu. (Guido Schuhmacher/sk)

```
programm : pseudo-irq        9e01 9f41
--------------------------------------
9e01 : ff 89 00 20 73 00 c9 21  6a
9e09 : d0 03 4c 6d 9e a5 cb c9  f5
9e11 : 40 d0 06 20 79 00 4c e7  d8
9e19 : a7 c9 04 f0 06 20 79 00  0b
9e21 : 4c e7 a7 a5 cb c9 40 d0  ad
9e29 : fa ad 01 9e f0 06 20 79  c1
9e31 : 00 4c e7 a7 18 a9 03 20  61
9e39 : fb a3 a5 7a 48 a5 7b 48  ef
9e41 : a5 39 48 a5 3a 48 a9 75  c1
9e49 : 48 8d 01 9e ad 02 9e 85  dc
9e51 : 14 ad 03 9e 85 15 20 a3  99
9e59 : a8 4c ae a7 a9 04 8d 08  c9
9e61 : 03 a9 9e 8d 09 03 a9 75  cc
9e69 : 8d 01 9e 60 20 73 00 c9  5c
9e71 : 46 f0 07 c9 4a f0 41 4c  f4
9e79 : f1 9e 20 73 00 c9 31 d0  e5
9e81 : 34 20 73 00 c9 4a d0 2d  2f
9e89 : 20 73 00 c9 55 d0 26 20  51
9e91 : 73 00 c9 4d d0 1f 20 73  8e
9e99 : 00 c9 50 d0 18 20 73 00  fc
9ea1 : 20 6b a9 a5 14 8d 02 9e  89
9ea9 : a5 15 8d 03 9e a9 00 8d  ef
9eb1 : 01 9e 4c ae a7 4c 08 af  47
9eb9 : 20 73 00 c9 42 d0 f6 20  93
9ec1 : 73 00 c9 41 d0 ef 20 73  c3
9ec9 : 00 c9 43 d0 e8 20 73 00  f6
9ed1 : c9 4b d0 e1 68 c9 75 d0  fc
9ed9 : 59 68 85 3a 68 85 39 68  77
9ee1 : 85 7b 68 85 7a a9 00 8d  ff
9ee9 : 01 9e 20 79 00 4c e7 a7  c2
9ef1 : c9 53 f0 07 c9 43 f0 1c  34
9ef9 : 4c 08 af 20 73 00 c9 46  24
9f01 : d0 b3 20 73 00 c9 31 d0  d6
9f09 : ac a9 75 8d 01 9e 20 73  05
9f11 : 00 4c ae a7 20 73 00 c9  09
9f19 : 46 d0 9a 20 73 00 c9 31  33
9f21 : d0 93 a9 00 8d 01 9e 20  c1
9f29 : 73 00 4c ae a7 4a 42 41  de
9f31 : 43 cb a9 2e 85 22 a9 9f  d9
9f39 : 85 23 4c 47 a4 01 00 a5  e9
```

Listing 11. Listing zu »Pseudo-Interrupt«

RESTORE für Unterprogramme

Will man in ein Programm ein schon vorhandenes Unterprogramm einfügen, kann es beim Lesen von DATAs Schwierigkeiten geben. Oft genügt die RESTORE-Anweisung nicht.

Wenn beide Programmteile, Hauptprogramm und Unterprogramme, DATAs enthalten, muß sichergestellt werden, daß auch wirklich die richtigen Werte gelesen werden. Wenn man nicht aufpaßt, kann es passieren, daß das Unterprogramm DATAs aus dem Hauptprogramm liest. Wie kann man das verhindern? Es gibt eine umständliche Methode: Man kann eine kleine Basic-Erweiterung einbauen, den RESTORE X-Befehl. Es geht aber auch einfacher. Die Zeropage, das sind die ersten 256 Byte des Speichers, hilft uns bei der Lösung des Problems. Genauer gesagt, die Adressen 65/66 und 122/123. Schlagen wir im C 64-Handbuch auf Seite 162 nach, dann steht dort:

65– 66 Adresse des aktuellen DATA-Elements
122–123 Basic-Zeiger innerhalb der Subroutine

Mit diesen Informationen läßt sich schon etwas anfangen. Wenn das Unterprogramm angesprungen wird, dann sollte der Zeiger in Speicherstelle 122/123 auf die Adresse des Unterprogramms im Speicher stehen. POKEt man diese Werte in die Zeilen 65/55 mit

```
POKE 65,PEEK(122)
POKE 66,PEEK(123),
```

so wird beim nächsten READ der Wert gelesen, der hinter dieser Basic-Zeile mit den POKEs steht, also das erste DATA-Element innerhalb des Unterprogramms. Nach dem Rücksprung aus dem Unterprogramm muß der Zeiger eventuell auch im Hauptprogramm wieder gestellt werden. In dem kurzen Demo-Listing (Listing 12) werden drei Unterprogramme in zufälliger Reihenfolge aufgerufen.

(Stephan Pätzold/sk)

```
1 REM     *************************        <128>
2 REM     *           DEMO          *      <010>
3 REM     *  SUBROUTINE-RESTORE     *      <071>
4 REM     *************************        <131>
5 PRINT"{CLR,6SPACE}TASTE DRUECKEN !"      <104>
6 PRINT:PRINT                              <114>
10 X=INT(RND(TI)*3)+1                      <125>
20 ON X GOSUB 1000,2000,3000               <040>
25 POKE 65,PEEK(122):POKE 66,PEEK(123)     <001>
30 READ A$:PRINT A$                        <066>
50 DATA " HAUPTPRG."                       <146>
100 GOTO 10                                <078>
1000 REM   *** SUBROUTINE 1 ***            <183>
1005 :                                     <042>
1010 POKE 65,PEEK(122):POKE 66,PEEK(123)   <221>
1020 FOR I=1 TO 4:READ A:PRINT A;:NEXT     <145>
1030 READ A$:PRINT A$;                     <105>
1040 POKE 198,0:WAIT 198,1                 <116>
1050 DATA 1,11,111,1111,"UP1{2SPACE}"      <085>
1060 RETURN                                <182>
1070 :                                     <108>
2000 REM   *** SUBROUTINE 2 ***            <164>
2005 :                                     <022>
2010 POKE 65,PEEK(122):POKE 66,PEEK(123)   <201>
2020 FOR I=1 TO 4:READ A:PRINT A;:NEXT     <125>
2030 READ A$:PRINT A$;                     <084>
2040 POKE 198,0:WAIT 198,1                 <095>
2050 DATA 2,22,222,2222,"UP 2 "            <076>
2060 RETURN                                <162>
2070 :                                     <088>
3000 REM   *** SUBROUTINE 3 ***            <145>
3005 :                                     <002>
3010 POKE 65,PEEK(122):POKE 66,PEEK(123)   <181>
3020 FOR I=1 TO 4:READ A:PRINT A;:NEXT     <105>
3030 READ A$:PRINT A$;                     <064>
3040 POKE 198,0:WAIT 198,1                 <075>
3050 DATA 3,33,333,3333,"UP{2SPACE}3"      <066>
3060 RETURN                                <141>
3070 :                                     <067>
```

Listing 12. Demoprogramm zu »RESTORE für Unterprogramme« mit zwei POKEs

RAM-Floppy

Wer kennt das nicht: ein paar Veränderungen an einem Programm – eine Zeile rein, eine andere raus – und nichts geht mehr. Das lästige Neuladen des Originalprogramms von Diskette können Sie ab jetzt vergessen.

Ist ein Programm mal wieder zu Tode editiert, werden Sie nun nicht mehr von den langen Ladezeiten der Floppy 1541 in Ihrem Programmierdrang gebremst. Mit »RAM-Floppy« (Listing 13) kann ein Programm bearbeitet werden, während man eine Kopie davon im RAM hat. In Sekundenschnelle kann die Kopie in den Basic-Speicher gebracht oder mit der Originalversion vertauscht werden. ganz einfach durch Eingabe von »@V« oder »@T«. Numerische Variablen bleiben dabei erhalten. Die »RAM-Floppy« besitzt eine Speicherkapazität von maximal 25 KByte. Der Speicher beginnt ab Adresse 40960.

Ein Problem ergibt sich im Speicherbereich des Kernel- und Basic-ROMs. Ein POKE-Befehl schreibt ins RAM, während die PEEK-Funktion auf das ROM zugreift. Noch komplizierter sieht es beim Zeichen-ROM und den I/O-Bausteinen aus. Wie Sie vielleicht aus unserem Grafikkurs wissen, gibt es in diesen Bereichen drei Speicheretagen. Der Inhalt der Zelle 1 regelt den Zugriff des Computers auf die verschiedenen Speicherebenen. Werden die Bits 0 und 1 in Adresse 1 gelöscht, sieht der Computer nur noch das RAM. Basic- und Kernel-ROM sind verschwunden. Löschen Sie diese Bits deshalb nur durch ein Maschinenspracheprogramm, wenn vorher Ein- und Ausgaben gesperrt wurden. Dies wird durch Setzen des Interruptregisters erreicht.

Das Maschinenprogramm besteht aus drei Teilen. Im Bereich von 40704 bis 40768 erfolgt die Auswertung der Befehle von »RAM-Floppy« und der Aufruf der beiden Unterprogramme, die das Tauschen oder Verschieben der Basic-Programme erledigen.

Das abgedruckte Basic-Programm (Listing 12) POKEt das Maschinenprogramm ab Adresse 40704. Mit SYS 40704 wird es initialisiert.

Die »RAM-Floppy« hat eine Kapazität von 25 KByte. Das Programm im Basic-Speicher kann zwar 38 KByte lang sein, läßt sich dann allerdings nicht mehr vollständig verschieben oder vertauschen. Der Speicherbedarf sollte auch bei Programmen mit vielen Variablen nicht außer acht gelassen werden. Bei langen Programmen mit vielen Variablen kann es durchaus vorkommen, daß die Programme zwar getauscht, die Variablen allerdings nicht mehr übernommen werden können.

(Uwe Klatt/sk)

```
0 REM************************************    <037>
1 REM*             RAM-FLOPPY           *    <029>
2 REM************************************    <039>
3 REM*  UWE KLATT                       *    <042>
4 REM*  BILLERBECKER STR. 27            *    <148>
5 REM*  4939 STEINHEIM                  *    <067>
6 REM*  TEL. 05233/5672                 *    <106>
7 REM************************************    <044>
8 POKE 53280,0:POKE 53281,11                 <021>
9 POKE 646,8                                 <072>
10 PRINT"BITTE WARTEN"                       <040>
11 REM ******************************        <023>
12 REM *** DATAS  LESEN            ***       <020>
13 REM ******************************        <025>
14 FOR I=40704 TO 40768:READ A:POKE I,A:S=
   S+A:NEXT                                  <100>
```

```
15 IF S<>6567 THEN END                           <181>
16 FOR I=40784 TO 40849:READ A:POKE I,A:S=
   S+A:NEXT                                      <143>
17 IF S<>14392 THEN END                          <098>
18 FOR I=40853 TO 40902:READ A:POKE I,A:S=
   S+A:NEXT                                      <169>
19 IF S<>20412 THEN END                          <147>
20 REM ******************************            <032>
21 REM ***          MENUE           ***          <197>
22 REM ******************************            <034>
23 PRINT CHR$(147)                               <052>
24 PRINT"{RVSON,40SPACE}";                       <194>
25 PRINT"{RVSON,SPACE}RAM-FLOPPY 25.5 KBYT
   E{18SPACE}";                                  <048>
26 PRINT"{RVSON,40SPACE}"                        <246>
27 PRINT"'@V'{2SPACE}VERSCHIEBT PROGRAMM I
   N{2SPACE}RAM-FLOPPY"                          <254>
28 PRINT"'@T'{2SPACE}VERTAUSCHT PROGRAMM M
   IT RAM-FLOPPY"                                <075>
29 REM ******************************            <041>
30 REM *** MC PROGRAMM STARTEN   ***             <222>
31 REM ******************************            <043>
32 SYS 40704:NEW                                 <211>
33 REM ******************************            <045>
34 REM *** DATAS FUER 1. MC TEIL ***             <097>
35 REM ******************************            <047>
36 DATA 169,159,133,56,133,52,169,0,133,55
   ,169,21,141,8,3,169                           <063>
37 DATA 159,141,9,3,96,32,115,0,240,4,201,
   64,240,3,76,231                               <111>
38 DATA 167,32,115,0,201,84,240,7,201,86,2
   40,12,76,8,175,32                             <255>
39 DATA 115,0,32,80,159,76,174,167,32,115,
   0,32,149,159,76,174                           <127>
40 DATA 167                                      <008>
41 REM ******************************            <053>
42 REM *** DATAS FUER 2. MC TEIL ***             <234>
43 REM ******************************            <055>
44 DATA 169,0,133,45,169,104,133,46,120,16
   5,1,41,252,133,1,169                          <051>
45 DATA 0,133,98,133,100,141,0,160,169,160
   ,133,101,169,8,133,99                         <246>
46 DATA 162,96,160,0,177,98,133,102,177,10
   0,145,98,165,102,145,100                      <094>
47 DATA 200,208,241,230,99,230,101,202,208
   ,232,165,1,9,3,133,1                          <062>
48 DATA 88,96                                    <112>
49 REM ******************************            <061>
50 REM *** DATAS FUER 3. MC TEIL ***             <114>
51 REM ******************************            <063>
52 DATA 120,165,1,41,252,133,1,169,0,133,9
   8,133,100,141,0,160                           <069>
53 DATA 169,160,133,101,169,8,133,99,162,9
   6,160,0,177,98,145,100                        <197>
54 DATA 200,208,249,230,99,230,101,202,208
   ,240,165,1,9,3,133,1                          <197>
55 DATA 88,96                                    <119>
```

Listing 13. Listing zu »RAM-Floppy«

Automatische Zeilennumerierung

Das lästige Durchnumerieren der Zeilen bei der Programmierung kann Ihnen dieses kleine Programm abnehmen.

Die Syntax des AUTO-Befehls ist:

←A anfangszeilennummer, schrittweite

Nach Eingabe dieses Befehls wird die Zeilennummer vorgegeben und nach RETURN um »schrittweite« erhöht.

Um aus dem AUTO-Modus wieder herauszukommen, muß man nach Vorgabe einer Zeilennummer »←« RETURN eingeben.

Falls man nach Vorgabe einer Zeilennummer die RETURN-Taste betätigt, wird die entsprechende Zeile, falls sie vorhanden ist, gelöscht. Hiermit lassen sich auch sehr schnell Programmblöcke löschen, falls man die RETURN-Taste gedrückt hält, die Zeilenvorgabe weiterläuft und die entsprechenden Zeilennummern gelöscht werden.

»←«=CHR$(95)
»A«=CHR$(65)

```
1010 REM********************************     <130>
1020 REM**      AUTO FUER C 64        **     <230>
1030 REM**           VON              **     <129>
1040 REM**      FRANK SIEDEL          **     <145>
1050 REM**      POSENER STR. 18       **     <101>
1060 REM**      2945 SANDE            **     <232>
1070 REM********************************     <192>
1080 :                                       <040>
1090 :                                       <050>
1100 :                                       <060>
1110 :DATA 169,11,141,8,3,169,192,141,9,3,
     96,32,115,0,8,201,95,240,4,40,76,231    <016>
1120 :DATA 167,32,115,0,201,65,208,245,32,
     115,0,24,32,107,169,165,20,133,38       <106>
1130 :DATA 165,21,133,39,32,253,174,24,32,
     107,169,165,20,133,40,165,21,133,41     <160>
1140 :DATA 169,129,141,2,3,169,192,141,3,3
     ,169,128,141,138,2,165,39,133,98,165    <110>
1150 :DATA 38,133,99,162,144,56,32,73,188,
     32,221,189,162,0,189,1,1,240,9,157      <044>
1160 :DATA 0,2,32,210,255,232,208,242,32,1
     8,225,201,95,240,30,201,13,240,45       <002>
1170 :DATA 157,0,2,232,32,98,165,76,134,16
     4,24,165,38,101,40,133,38,165,39,101    <017>
1180 :DATA 41,133,39,76,75,192,169,131,141
     ,2,3,169,164,141,3,3,169,0,141,138      <181>
1190 :DATA 2,40,76,116,164,32,118,165,76,1
     34,164                                  <239>
1200 :                                       <160>
1210 :                                       <170>
1220 PRINT CHR$(147)                         <233>
1230 SU=0                                    <254>
1240 FOR I=1 TO 170                          <217>
1250 READ A                                  <018>
1260 SU=SU+A                                 <107>
1270 POKE 49151+I,A                          <145>
1280 NEXT                                    <020>
1290 IF SU<>17417 THEN PRINT "FEHLER IN DE
     N DATAZEILEN":END                       <228>
1300 PRINT"OK":END              © 64'er      <023>
```

Listing 14. Listing zu »Automatische Zeilennumerierung«

Das Programm (Listing 14) als Basic-Lader eintippen, anschließend mit RUN starten. Falls »FEHLER IN DEN DATAZEILEN« erscheint, DATAs auf Tippfehler überprüfen. Falls »OK«, kann die Basic-Erweiterung mit SYS 49152 initialisiert werden. Nun hat man das Basic um den Befehl »A« erweitert.

(Frank Siedel/sk)

INPUT mit Komma

Diese INPUT-Routine von Jürgen Reinert ersetzt den INPUT-Befehl. Sie erlaubt Komma, Doppelpunkt und Strichpunkt als zusätzliche Satzzeichen bei der Eingabe. Sonst funktioniert sie genauso wie der INPUT-Befehl. Die Routine übernimmt alle Zeichen der Tastatur, auch führende Leerzeichen (Leerzeichen vor Beginn des Textes).

Die Variable AA enthält die aktuelle Eingabe in ASCII-Code. II bildet die Laufvariable für Schleifen und XX$ enthält den eingegebenen Text.

Das »Herz« dieses Einzeilers ist die Eingaberoutine ab Adresse 42336. Diese schreibt alle 80 Zeichen einer Bildschirmzeile in den Basic-Eingabepuffer, der bei Adresse 512 beginnt. Dann liest das Programm Zeichen für Zeichen den Eingabepuffer bis zur genannten 0 und stellt daraus den String XX$ zusammen.

Erfolgt keine Eingabe, das heißt nur die RETURN-Taste wurde gedrückt, so wird die Routine mit XX$=CHR$(32) verlassen. In allen anderen Fällen enthält XX$ alle sichtbaren, eingegebenen Zeichen (außer Steuerzeichen).

```
1 SYS 42336:XX$="":FOR II=512 TO 600:AA=PE
  EK(II):IF AA THEN XX$=XX$+CHR$(AA):NEXT
10 :
20 REM INPUT
```

Zeilen löschen am Bildschirm

Diese kleine Routine von Stefan Keimeier löscht bestimmte Zeilen auf dem Bildschirm. Dabei wird eine Maschinenroutine des C 64 verwendet, die diejenige Zeile vom Bildschirm löscht, deren Zeilennummer im X-Register steht. Die Zeilennummer wird zunächst in die Speicherzelle gePOKEt, deren Inhalt der SYS-Befehl in das X-Register übernimmt. Dabei wird von 0 bis 24 gezählt.

Die Variable LN enthält die Zeilennummer (0 bis 24), V gibt die Von-Zeile und B die Bis-Zeile an (gelöscht wird »von« »bis«).

```
10 FOR LN=V TO B:POKE 781,LN:SYS 59903:NEXT
20 :
30 REM FUER NUR EINE ZU LOESCHENDE ZEILEGILT:
40 :
50 POKE 780,LN:SYS 59903
60 :
70 REM                        STEFAN KEIMEIER
```

Grafikbildschirm löschen mit dem DIM-Befehl

Die Variablen A und B müssen vor der DIM-Anweisung angelegt werden:
(A=0:B=0), da im Programmverlauf das Variablenfeld beeinflußt wird.
Dann werden durch
A=PEEK(49):B=PEEK(50) die Werte für das Variablenende gesichert. Durch das Dimensionieren einer Variablen wird ein entsprechend großer Platz hinter dem Variablenende auf Null gesetzt.

Der Bildschirmspeicher fängt an bei Adresse 8192 und geht bis Adresse 16191. Hinter dem bisherigen Variablenende (Adresse: A+B*256) wird die Variable F (auf der Adresse 16191-a-b*256) angelegt. Pro indizierter Variablen werden 5 Byte freigemacht. Dabei wird die mit 0 indizierte Variable nicht berücksichtigt, ebenso die ersten 7 Byte für Variablenname und Dimension.

Durch POKE 49,A:POKE 50,B werden die alten Werte wiederhergestellt. F ist nun nicht mehr dimensioniert.

Wenn das Variablenende größer oder gleich 8186 ist, wenn also das Programm einschließlich Variablenfeld größer ist als 6 Byte, wird der Bildschirm nicht ganz gelöscht.

Soll das Grafikprogramm zum Speichern von Bildern benutzt werden, darf man nicht vergessen, den Zeiger für Speichergrenzen (PEEK(55)+PEEK(56)*256) auf eine Adresse kurz hinter den Bildschirmspeicher zu setzen.

Der Autor des Programms, Manfred Hedtke, schlägt vor, im Grafikprogramm die Zeile 10 immer dann aufzurufen, wenn über die Tastatur »SHIFT + CTRL HOME« (entsprechend »freier Bildschirm im Textmodus«) eingegeben wird.

```
10 a=0:b=0:a=peek(49):b=peek(50):dimf((1
6191-a-b*256)/5):poke49,a:poke50,b
20 rem
```

Ein grafischer Disassembler: DI-AS

Der Speicherbereich des C 64 läßt sich in 256 Seiten à 256 Byte aufteilen. DI-AS interpretiert die Speicherinhalte als Bildschirmcode und stellt die 64 KByte des C 64 Seite für Seite auf den Zeilen 7–13 des Bildschirms dar. Die Seitennummer wird am oberen Bildschirmrand angezeigt. Andreas Carl, der Autor dieses Programms, sagt zur Bedienung des Dissassemblers:

Bevor das Progamm geladen und gestartet wird, muß mit »SYS64738« unbedingt ein Reset durchgeführt werden. Dann sollte der Bildschirm mit »CLR« gelöscht werden. Mit »RUN« wird das Programm gestartet. Dann stehen dem Benutzer die folgenden Optionen offen:

- im Speicher vorwärtsblättern <CURSOR-rechts>
- im Speicher rückwärtsblättern <CURSOR-abwärts>
- die Seite NR betrachten <R/S POKE3,NR:CLR:RUN>

Es stehen umfangreiche Fehlerbehandlungen zur Verfügung.

```
5 print"1bq(HP :"peek(3)" ":pok
e41,5:geta$:poke3,peek(3)-(a$="")+(a$="
"):sys1024:goto5
```

Speicherblockverschiebung

Dieser Einzeiler von Jens Baas dient zur Übertragung von Speicherblöcken. Er ist zum Beispiel nützlich, wenn Sie das Basic oder das Betriebssystem vom ROM ins RAM verlegen wollen. Die Variablen mit »L« bezeichnen jeweils das Low-Byte der Adresse, die Variablen mit »H« das High-Byte. Für eine Adresse X lassen sich diese so berechnen:
AL=X-256*INT(X/256):AH=INT(X/256).

Die Routine benutzt die Blockverschiebe-Routine des ROMs.

```
1 poke95,al:poke96,ah:poke90,el:poke91,e
h:poke88,nl:poke89,nh:sys41919
11 rem beispiel:
12 poke95,0:poke96,160:poke90,0:poke91,1
92:poke88,0:poke89,192:sys41919
13 rem ↑↑ basic ins ram ↑↑
```

Ein einfaches Renumber

Diese Routine von Georg Wichert numeriert ein Basic-Programm, das bis zu 255 Zeilen hat, in wenigen Sekunden neu. Die erste Zeilennummer ist dann 0, die Schrittweite ist 1, Sprungadressen der Befehle GOTO und GOSUB bleiben beim Neu-Numerieren unverändert. Jedes Basic-Programm belegt die Speicherplätze 2048 bis PEEK(45)+PEEK(46)*256-3. PEEK(2049)+PEEK(2050)*256 gibt an, bei welcher Adresse die erste Zeile aufhört. Die erste Zeilennummer A wird aufgrund der Adressen 2051 und 2052 folgendermaßen berechnet:
A=PEEK(2051)+PEEK(2052)*256.
Die zweite Zeilennummer findet man im Speicher an den Adressen
PEEK(2049+PEEK(2050)*256+2 und
PEEK(2049)+PEEK(2050)*256+3.

Die Zeilennummern befinden sich jeweils an den Adressen A+2 (Low Byte) und A+3 (High Byte).

Ist das Basic-Programm, das neu numeriert wird, länger als 255 Zeilen, dann muß der Einzeiler zum Zweizeiler abgeändert werden. Dann wird folgendes wichtig: Aus Platzgründen wird in dieser Programmversion an die Stelle A+2 der Wert z gePOKEt (z=0,1,2...,n-1; n-1 steht für die Anzahl der Zeilen des Basic-Programms, das umnumeriert werden soll, inklusive dem Einzeiler) und an die Stelle A+3 den Wert 0.

```
1 FOR A=2049 TO PEEK(45)+PEEK(46)*256-3:PO
  KE A+2,Z:POKE A+3,0:A=PEEK(A)+PEEK(A+1)*
  256-1:Z=Z+1:NEXT
10 :
20 REM RENUMBER (GEORG WICHERT)
```

Der Befehl Goto X – eine Basic-Erweiterung

Dieses Programm von Reinhard Jurk simuliert den im Commodore-Basic nicht vorhandenen Befehl GOTO X.

Zu dem etwas seltsamen Aussehen der Programmzeile: In den Anführungszeichen in Zeile 10 steht ein Maschinenprogramm (Listing 15), dessen Opcodes im Listing des Basic-Interpreters diese merkwürdigen Zeichen erzeugen.

Zur Eingabe:
In diesem DATA-Lader (Listing 16) ist ein Generierungsprogramm für diese Zeile 10 angegeben. Dieses wird eingegeben und gestartet. Hat man sich vergewissert, daß das Programm richtig eingegeben wurde, gibt man »LIST« ein, gefolgt von »NEW«. Dann fährt man mit dem Cursor auf Zeile 10 und drückt RETURN. Die Routine steht nun im Speicher.

Zum Einbinden der Routine in eigene Programme ändert man die Zeilennummern entsprechend ab, weist der Variablen LL% den Wert der anzuspringenden Zeile zu und springt mit »GOTO« zu der Zeile, in der GOTO X steht.

```
10 syspeek(61)+256*peek(62)+26:"[illegible]"
20 rem
30 rem goto x: ll%=zeilennummer:goto10
40 rem
```

```
  BITTE UNBEDINGT GENAUSO EINGEBEN !!!
  43 KLAMMERAFFEN (@)
5 GOTO20
10 SYSPEEK(61)+256*PEEK(62)+26:"@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
20 FORA=2082TO2124:READX:S=S+X:POKEA,X:N
EXT:IFS<>5185THENPRINT"DATA ERROR !":END
30 PRINT"OK."
40 DATA169,204,133,69,133,70,165,157,133
,85,133,86,160,12,200,145,85,162,129
50 DATA202,200,138,145,85,172,128,163,16
6,46,165,45,32,17,177,32,43,175,32,1
60 DATA184,76,163,168
READY.
```

Listing 16. Der Einzeiler als DATA-Lader

```
C*
     PC  IRQ  NV-BDIZC AC XR YR SP
.;  E147 EA31 00110001 08 08 02 F6

.,  081A  A9 CC       LDA #$CC
.,  081C  85 45       STA $45
.,  081E  85 46       STA $46
.,  0820  A5 9D       LDA $9D
.,  0822  85 55       STA $55
.,  0824  85 56       STA $56
.,  0826  A0 0C       LDY #$0C
.,  0828  C8          INY
.,  0829  91 55       STA ($55),Y
.,  082B  A2 81       LDX #$81
.,  082D  CA          DEX
.,  082E  C8          INY
.,  082F  8A          TXA
.,  0830  91 55       STA ($55),Y
.,  0832  AC 80 A3    LDY $A380
.,  0835  A6 2E       LDX $2E
.,  0837  A5 2D       LDA $2D
.,  0839  20 11 B1    JSR $B111
.,  083C  20 2B AF    JSR $AF2B
.,  083F  20 01 B8    JSR $B801
.,  0842  4C A3 A8    JMP $A8A3
```

Listing 15. Die Bedeutung der DATAs im Assembler-Code

Trick 17 mit ON ... GOTO

Eine sehr interessante Version einer ON...GOTO-Anweisung stammt von Peter Zankl (Listing 17).

Der Ausdruck (A$= "A") hat den Wert -1, wenn ein A eingegeben wurde. Sonst hat er den Wert 0. Rechnen Sie nach oder probieren Sie es aus!

```
100 rem tastaturabfrage mit sprung
200 rem :
300 rem vorher:
400 :
410 geta$:ifa$=""then410
420 ifa$="a"then2000:rem programmteil a
430 ifa$="b"then3000:rem programmteil b
440 ifa$="x"then end:rem ende
450 goto 410
499 :
500 rem nachher:
600 :
610 geta$:on1-(a$="a")-2*(a$="b")-3*(a$=
"x")goto610,2000,3000:end
620 :
```

Listing 17. Tastaturabfrage

List-Schutz für Basic-Programme

Dieser List-Schutz (Listing 18) ist für Nichteingeweihte sehr verblüffend. Die Grundidee dazu stammt aus dem Bericht »Disketten-Manipulationen« aus der 64'er, Ausgabe 6/85. Er wurde jedoch etwas ausgebaut, so daß hier beim Listen alle Steuercodes aktiv werden. Dies wird dadurch erreicht, daß man in eine Speicherstelle vor den Codes die Zahl 141 schreibt. Um nun ein Programm zu schützen, lädt man es und gibt zwei Zeilen ein. (Listing 18)

Danach gibt man im Direktmodus ein:

```
POKE 2067,71 : POKE 2073,141 : POKE 2118,0 :
POKE 2119,0
```

Listet man nun das Programm, so wird der Bildschirm gelöscht und der Text in der REM-Zeile ausgegeben. Durch das künstlich erzeugte Basic-Programm-Endezeichen (drei Nullen) wird das Listen abgebrochen. Wird im Programm dann auch noch durch `POKE 788,52` die RUN-STOP-, und durch `POKE 792,193` die RESTORE-Taste ausgeschaltet, kann keiner mehr an das Programm. Aufheben läßt sich dieser List-Schutz nur mit einem Monitor und mit der Kenntnis der Funktionsweise. (Thomas Uttendorfer/sk)

```
1 POKE 2067,73:GOTO 10                                  <134>
2 REM"A{CLR,3DOWN,YELLOW}IT IS NOT ALLOWED
  TO LIST THIS PROGRAM{BLUE}AA                          <123>
```

© 64'er

Listing 18. Listing zu »List-Schutz für Basic-Programme«

Listhilfe

Wenn man nach Eingabe des LIST-Befehls das LISTen will, muß man «RUN/STOP» betätigen. Für weiteres LISTen muß erneut der entsprechende Befehl gegeben werden. Das Programm »SHIFT-HOLD« (Listing 19) bringt hier Abhilfe: Durch Drücken der SHIFT-Taste kann man das Listing anhalten; läßt man sie wieder los, wird weiter gelistet.

(Oliver Hertel/U.Beenen)

```
Name : shift-hold              c000 c021
----------------------------------------
c000 : a9 0b 8d 14 03 a9 c0 8d  b0
c008 : 15 03 60 ad 8d 02 c9 01  7f
c010 : f0 03 4c 31 ea a5 cb c9  5a
c018 : 40 d0 f7 20 87 ea 4c 0b  d9
c020 : c0 da 32 f5 fa fc b9 5f  d6
```

Listing 19. »SHIFT-HOLD«. Bitte mit dem MSE (Seite 159) eingeben.

List- und Löschschutz leichtgemacht

Es wurden schon viele Methoden veröffentlicht, um ein Basic-Programm gegen Listen zu schützen. Aber alle mir bekannten Möglichkeiten weisen entschiedene Nachteile auf. Entweder der Schutz ist nicht sicher genug und leicht zu entfernen, oder er ist viel zu aufwendig.

Ich habe mich daher entschlossen, ein Programm zu schreiben, das diese Mängel umgeht und sogar noch andere positive Merkmale aufweist.

Zunächst eine Zusammenfassung von drei mir bekannten Listschutzmöglichkeiten mit ihren Vor- und Nachteilen:

1. Möglichkeit

In die erste Zeile des Basic-Programms (zum Beispiel Zeilennummer 1) wird REM, gefolgt von zwei Anführungszeichen und SHIFT L, geschrieben.

`1 REM""L (RETURN)`

Der Cursor wird nun auf das zweite Anführungszeichen gesetzt und sechsmal SHIFT INST gedrückt (das Anführungszeichen wird um sechs Positionen nach rechts geschoben). Dann wird sechsmal DEL eingegeben (es erscheinen als Steuerzeichen sechs reverse T) und die Zeile mit (RETURN) gespeichert. Wird nun der LIST-Befehl aufgerufen, meldet sich der Rechner mit:

"

?SYNTAX ERROR
READY.

Auf den ersten Blick sehr beeindruckend, aber durch Entfernen dieser Zeile ist der Listschutz wieder aufgehoben. Außerdem ist ein »LIST 2« noch möglich.

2. Möglichkeit

In jede Basic-Zeile werden synthetische Steuerzeichen eingefügt (genaue Beschreibung im 64'er-Magazin, Ausgabe 6/84). Diese Methode ist zwar recht sicher, will man aber alle Zeilen eines längeren Basic-Programms schützen, ist der Aufwand viel zu groß, vom Speicherplatzbedarf der Steuerzeichen ganz abgesehen.

3. Möglichkeit

Durch `POKE 775,200` wird der Listbefehl außer Kraft gesetzt, durch `POKE 775,167` wird diese Wirkung wieder aufgehoben. Dieser Listschutz ist zwar wirkungsvoll, aber er muß erst durch diesen POKE-Befehl aktiviert werden. Nach dem Laden eines Programms ist er daher noch nicht aktiv.

Das hier vorgestellte Programm erzeugt nicht nur einen sicheren Listschutz, sondern schützt auch vor dem Löschen einzelner Basic-Zeilen. So können zum Beispiel Hinweise auf ein Kopierrecht und auf den Autor eines Programms nicht geändert oder entfernt werden. Auch kann ein so gesichertes Programm nur mit RUN gestartet werden, ein RUN, gefolgt von einer Zeilennummer, führt zu einer Fehlermeldung. Jede Zeile des Programms ist geschützt, es können also auch einzelne Zeilen nicht gelistet werden. Einzige Bedingung für die Verwendung dieses Schutzes: Das zu schützende Programm darf keine Zeilennummern 0 und 1 enthalten. Ansonsten wird eine Fehlermeldung ausgegeben und das Programm bleibt unverändert.

Das Listschutzprogramm (Listing 20) liegt als Basic-Lader vor. Nachdem es richtig abgetippt wurde, kann es durch RUN gestartet werden. Das Maschinenprogramm steht dann im Speicher ab der Adresse 50000 zur Verfügung. Das zu schützende Basic-Programm kann nun geladen werden, durch SYS 50000 wird das Schutzprogramm aktiviert und das Basic-Programm geschützt. Es kann nun

```
0 REM ************************************      <131>
1 REM *        (C) U. V. GAISBERG        *      <195>
2 REM *         AM ZUCKERBERG 70         *      <080>
3 REM *         7140 LUDWIGSBURG         *      <233>
4 REM *         TEL. 07141/55910         *      <056>
5 REM ************************************      <136>
6 FOR I=0 TO 340:READ A:B=B+A:POKE 50000+I
  ,A                                            <099>
7 NEXT I                                        <091>
8 IF B <> 33527 THEN PRINT"FEHLER IN DATAS
  !":END                                        <042>
9 PRINT"OK !":END                               <113>
10 REM DATAS FUER MASCHINENPROGRAMM             <157>
11 DATA 169,0,141,32,208,141,33,208,169,1,
   141,134,2,32,68,229,174,3,8,172              <018>
12 DATA 4,8,192,0,208,7,224,2,176,3,76,206
   ,195,162,0,142,134,2,169,32,32               <221>
13 DATA 210,255,232,224,50,208,246,162,0,1
   89,21,196,157,0,4,232,224,29,208             <001>
14 DATA 245,169,24,157,0,4,232,224,69,208,
   246,162,0,189,50,196,157,80,4,232            <190>
15 DATA 224,8,208,245,162,0,189,58,196,157
   ,120,4,232,224,8,208,245,162,10              <073>
16 DATA 160,0,24,32,240,255,169,19,162,13,
   160,4,141,119,2,142,120,2,142,121            <061>
17 DATA 2,142,122,2,132,198,96,162,0,189,9
   9,196,32,210,255,232,224,31,208              <065>
18 DATA 245,96,32,68,229,162,10,160,0,24,3
   2,240,255,162,1,142,134,2,202,189            <210>
19 DATA 66,196,32,210,255,232,224,33,208,2
   45,169,20,162,17,160,255,141,18              <250>
20 DATA 8,142,29,8,140,4,8,162,0,189,130,1
   96,157,32,8,232,224,34,208,245               <171>
21 DATA 96,48,18,5,13,34,148,148,148,148,1
   48,148,148,148,148,148,34,12,12              <122>
22 DATA 9,19,20,19,3,8,21,20,26,26,76,49,1
   9,25,19,50,48,57,56,19,25,19,53              <045>
23 DATA 48,49,52,48,80,82,79,71,82,65,77,7
   7,32,45,32,40,67,41,32,85,46,86              <224>
24 DATA 46,71,65,73,83,66,69,82,71,32,32,4
   9,57,56,52,66,73,84,84,69,32,90              <032>
25 DATA 69,73,76,69,32,48,32,85,78,68,32,4
   9,32,69,78,84,70,69,82,78,69,78              <005>
26 DATA 32,33,169,255,141,4,8,169,131,162,
   164,141,2,3,142,3,3,76,131,164               <172>
27 DATA 165,2,141,4,8,169,32,162,8,141,2,3
   ,142,3,3,96,0                                <150>
```

© 64'er

Listing 20. Listing zu »List- und Löschschutz leichtgemacht«

wieder auf Kassette/Diskette gespeichert werden. Das mit dem Listschutz versehene Programm ist nur um wenige Bytes größer als vorher.

Funktionsweise

Das Maschinenprogramm generiert zwei Basic-Zeilen mit den Zeilennummern 0 und 1. Die Zeile 0 ist eine REM-Zeile, in der ein unlistbares Zeichen (SHIFT L) steht. Hinter diesem Zeichen stehen dann noch zwei kurze Maschinenprogramme, deren Funktionen im folgenden noch erklärt werden. In der zweiten Zeile steht ein SYS-Befehl, der eine der beiden Maschinenroutinen in Zeile 0 startet. Sind diese beiden Zeilen nun erzeugt, wird die Zeilennummer 0 durch eine höhere, eigentlich unerlaubte Zeilennummer (größer 64000) ersetzt. Diese Zeile kann daher auch nicht gelöscht werden.

Da alle nun folgenden Zeilen des Programms kleiner sind als die erste, können diese vom Computer nicht mehr erkannt werden. Ein Sprung in eine solche Zeile führt zu der Fehlermeldung: ?UNDEF'D STATEMENT ERROR. Es kann daher auch keine Zeile gelöscht werden, da diese für den Computer ja nicht mehr vorhanden sind.

Der einzige Nachteil ist, daß es nicht nur ein perfekter List- und Löschschutz, sondern auch ein RUN-Schutz ist (auch Sprungziele innerhalb des Programms können nicht gefunden werden).

Wird das geschützte Programm gestartet, trifft der Interpreter als erstes auf den SYS-Befehl in Zeile 1. Es folgt ein Sprung in das Maschinenprogramm in der REM-Zeile. Dort wird die Zeilennummer wieder auf 0 gesetzt, und der Vektor auf den Basic-Warmstart wird auf die zweite Maschinenroutine gesetzt.

Nun kann das Basic-Programm ohne Fehler ausgeführt werden. Wird der Programmlauf unterbrochen (durch STOP-Taste, Fehlermeldungen, Programmende und so weiter), wird das zweite Maschinenprogramm über den Basic-Warmstartvektor angesprungen. Dort wird die Zeilennummer wieder hochgesetzt, der Warmstartvektor wieder auf den normalen Wert gebracht und die Warmstartroutine angesprungen. Das Programm liegt nun wieder in der geschützten Form vor. (Ulrich von Gaisberg/sk)

Eine Zeile = kompletter Datenschutz

Diese Routine von Volker Ritzhaupt erreicht, daß auf einer Diskette weder das Inhaltsverzeichnis gelistet, Programme gelöscht noch irgendetwas auf die Diskette geschrieben werden kann. Das Laden der Programme funktioniert hingegen ganz normal. Der Inhalt der Diskette kann nur durch Formatieren mit einer neuen ID-Nummer noch manipuliert werden.

Das Directory wird nicht mehr gelistet, weil es als Basic-Programm geladen wird und in dem veränderten Directory drei Nullen am Anfang (für den Interpreter das Zeichen für Programmende) erscheinen. Das »Directory-Programm« endet also schon nach 10 Byte. Die ersten 5 Byte stellen den Zeilenanfang und die Zeilennummer dar. Darauf folgen, wie immer am Anfang eines Inhaltsverzeichnisses, ein Leerzeichen und ein Anführungszeichen. Um diese drei Zeichen beim Listen verschwinden zu lassen, folgen nun drei CHR$(20), also drei DELETEs. Dies bewirkt, daß das Directory beim Listen völlig verschwindet. Da nun unmittelbar die drei Nullen folgen, wird das Listen hier abgebrochen. Auf den LIST-Befehl erscheint nur die Meldung »READY«.

```
1 open1,8,3,"#":open2,8,15,"b-p3,144":pr
int#1,"[illegible]"chr$(0)chr$(0)chr$(0):print#2
,"u2:3,0,18":print#2,"i
2 rem
3 rem verhindert auflisten des directory
      und macht schreibschutz auf disk
4 rem
```

Reset-Helfer für C 64

Das Betriebssystem des C64 enthält ab der Adresse $FD02 ein Unterprogramm, das im Steckmodulbereich ab $8000 nach der Zeichenfolge »CBM80« sucht. Nach dem Einschalten des Computers oder nach einem Reset wird dieses Unterprogramm jedesmal aufgerufen. Werden ab der Adresse $8003 die Zeichen »CBM80« gefunden, dann wird nicht zum Basic-Start gesprungen, sondern das Betriebssystem nimmt an, daß ein Modul eingesteckt ist, holt sich aus der Speicherzelle $8000/$8001 die Startadresse des Modulprogramms und verzweigt dorthin.

Das kleine »Reset-Helfer«-Programm (Listing 21) nutzt dies aus, indem es die genannten Speicherstellen in geeigneter Weise abändert. Es schreibt die »CBM80«-Zeichenfolge ab $8003 ins RAM und läßt die Speicherstellen $8000/ $8001 auf den Basic-Warmstart zeigen.

Wenn man jetzt einen Reset auslöst, kommt man wie gewohnt aus allen »abgestürzten« Programmen heraus, ein vorhandenes Basic-Programm bleibt aber erhalten.

(Henning Zipf/sk)

```
100 REM *** RESET-HELFER ***              <141>
110 REM                                   <172>
120 REM HENNING ZIPF                      <168>
130 REM KIRCHSTR. 8                       <036>
140 REM 6086 RIEDSTADT 5                  <150>
150 REM TEL. (06158) 72453                <243>
160 REM                                   <222>
170 FOR I=1 TO 9                          <016>
180 READ D                                <244>
190 POKE 32767+1,D                        <196>
200 NEXT I                                <028>
210 POKE 53280,14:POKE 53281,6            <166>
220 PRINT CHR$(147);CHR$(5)               <071>
230 PRINT" C 64 CHANGED RESET VEKTOR"     <067>
240 PRINT                                 <086>
250 PRINT" 64 K RAM SYSTEM 38911 BASIC BYT
    ES FREE"                              <233>
260 PRINT                                 <108>
270 NEW                                   <154>
280 DATA 0,0,255,0,195,194,205,56,48      <015>
290 REM ERST SAVE, DANN RUN !             <160>
```

Listing 21. Listing zu »Reset-Helfer für C64«

Das elektronische Tagebuch

Uralt schon ist das Bestreben vieler Leute, ihre geheimsten und privatesten Aufzeichnungen so gut wie möglich vor allzu neugierigen Mitmenschen zu schützen. Und wer kennt sie nicht, die Kladde mit dem Schlößchen, der solche Geheimnisse anvertraut werden – mit dem blauäugigen Glauben, es gäbe niemanden, der auf die Idee käme, sich dem Schloß mit einer aufgebogenen Büroklammer zu nähern...

Es liegt also nahe, sich nach einem wirkungsvolleren Schutz umzusehen. Der Computer bietet sich für solch verantwortungsvolle Dienste geradezu an. Es fehlt also nur noch das passende Programm.

Das »elektronische Tagebuch« in seiner Mikroausführung (Listing 22) ist da genau richtig. Die Vorteile liegen auf

der Hand: Das Programm ist sehr kurz (eine Bildschirmseite) und zugleich recht komfortabel. Seine Benutzung bereitet also weder bei der Eingabe noch bei der Anwendung großen Aufwand.

Das Programm läßt sich theoretisch auf jedem Commodore-Computer verwenden, einzige Voraussetzung ist natürlich ein externes Speichermedium, sprich: Datasette oder Floppy.

Die vorliegende Version ist für den Gebrauch mit dem C64 und Diskettenstation vorgesehen, die Anleitung zum Umschreiben auf Kassettenbetrieb und Hinweise für Benutzer anderer Computer folgen weiter unten.

Das Programm ist im Vergleich zur beschriebenen Kladde recht sicher. Absolut unknackbar ist es zwar nicht, aber es gehört Intelligenz statt Fingerspitzengefühl dazu, aus einer polyalphabetisch verschlüsselten und zu scheinbar beziehungslosen Zahlenfolgen umgewandelten Buchstabenkombination ein lesbares Satzgefüge (sprich: Text) zu machen. Außerdem steht es jedem frei, seine eigene Chiffriermethode zu entwickeln.

Doch kommen wir zum Programm selbst. Der Aufbau ist denkbar einfach: Gleich nach dem Start wird aus Sicherheitsgründen (für Ihren Text) die RUN/STOP-RESTORE-Funktion ausgeschaltet. In dem jetzt erscheinenden Mini-Menü müssen Sie sich entscheiden, ob Sie einen Text eingeben und codieren oder ob Sie ihn decodieren und lesen wollen. Dann werden Sie daran erinnert, eine Diskette (beziehungsweise Kassette) ins Laufwerk zu legen.

Sobald das alles erledigt ist, müssen Sie Ihre persönlichen Codezahlen eingeben, mit denen der Text »bearbeitet« werden soll. Für diese Codezahlen ist keine Begrenzung vorgesehen, da der Computer jedoch mit ihnen rechnen muß, empfiehlt es sich, ein manierliches Maß beizubehalten. (Mit Zahlen im dreistelligen Bereich sollte es keine Probleme geben.)

```
5 PRINT"(CTRL-N,CLR,2DOWN,3SPACE)ELEKTRONI
  SCHES TAGEBUCH"                           <009>
10 POKE 808,254:PRINT"(2DOWN,3SPACE)MENUE:
   ":INPUT"(DOWN,3SPACE)1=CODIEREN, 2=DECO
   DIEREN";W                                <045>
12 PRINT"(DOWN,3SPACE)BITTE DISKETTE EINLE
   GEN!(DOWN)"                              <105>
15 PRINT TAB(26)"TT.MM.JJJJ(2UP)":INPUT"(D
   OWN,18SPACE)DATUM ";D$                   <191>
18 PRINT"(DOWN,3SPACE)CODEZAHLEN X,Y,Z":IN
   PUT"(DOWN,3SPACE)DURCH KOMMA GETRENNT "
   ;A,B,C                                   <066>
20 DIM T(839),T$(839):X=0:IF W=2 THEN 70    <127>
25 PRINT"(CLR,DOWN,3SPACE)TEXTEINGABE (MAX
   . 839 ZEICHEN)":PRINT"(3SPACE)'*' = END
   E":PRINT"(DOWN)>";                       <126>
30 GET T$(X):IF T$(X)=""THEN 30             <202>
35 IF T$(X)="*"OR X=839 THEN 50             <096>
40 PRINT T$(X);:X=X+1:GOTO 30               <184>
50 OPEN 1,8,2,D$+",S,W":PRINT#1,X:FOR I=0
   TO X-1:B=B+(A*C):PRINT#1,ASC(T$(I))+C+B <218>
60 NEXT I:CLOSE 1:GOTO 99                   <143>
70 PRINT"(CLR,DOWN)>";:OPEN 1,8,2,D$+",S,R
   ":INPUT#1,X:FOR I=0 TO X-1:B=B+(A*C):IN
   PUT#1,T(I)                               <019>
80 T(I)=(T(I)-B)-C:IF T(I)<0 OR T(I)>255 T
   HEN CLOSE 1:PRINT"(DOWN)FALSCHE EINGABE
   ":GOTO 99                                <047>
90 PRINT CHR$(T(I));:NEXT I:CLOSE 1         <080>
99 CLR:POKE 808,237:END:REM -PTS,1985-      <004>
```


Listing 22. »Elektronisches Tagebuch«. Niemand hat Zutritt zu Ihren privaten Aufzeichnungen!

Jetzt teilt sich der Weg. Haben Sie vorhin im Minimenü eine 1 (für »codieren«) getippt, so können Sie jetzt Ihren Text eingeben. Dimensioniert ist ein Feld von 839 Zeichen, also gerade soviel, daß der Anweisungstext noch auf dem Bildschirm sichtbar bleibt und auch später, beim Decodieren, nichts vom Bildschirm nach oben »wegrutscht«. Wenn Sie übrigens Textteile mit »INST/DEL« löschen, so werden *alle* eingegebenen Zeichen, also sowohl die ursprünglichen Buchstaben als auch jedes »INST/DEL« mitgezählt und dann mitverschlüsselt. Damit lassen sich beim späteren Decodieren recht reizvolle Effekte erzielen. Probieren Sie's doch einfach mal aus!

Selbstverständlich können Sie die Texteingabe auch schon vor Erreichen des 839. Zeichens abschließen: Sie brauchen nur die Sterntaste (»*«) zu drücken. Wenn Sie den Stern in Ihren Texten verwenden wollen, setzen Sie an der entsprechenden Programmstelle (Zeile 35) einfach nur ein anderes ENDE-Zeichen ein.

Sobald die Eingabe beendet ist, schreibt der Computer zunächst die Anzahl der Zeichen in die Datei. Dann errechnet er Buchstabe für Buchstabe nach einem bestimmten, sich ständig systematisch verändernden Prinzip (Sie finden es in Zeile 50) eine bestimmte Zahl und speichert diese ebenfalls auf der Diskette (oder auf der Kassette).

Anschließend, und hier vereinigen sich die beiden Programmzweige wieder (Zeile 99), werden durch »CLR« alle Variablen, also der Text und Ihre Codezahlen, gelöscht. Nun wird die Blockierung von RUN/STOP und RESTORE aufgehoben und das Programm beendet.

Der zweite Zweig, das Decodieren, beginnt in Zeile 70. Hier wird genau umgekehrt verfahren wie im ersten Teil: Erst wird die Anzahl der zu decodierenden Zahlen gelesen, dann kommen die Zahlen selbst an die Reihe. Stück für Stück werden sie aus der Datei geholt, entschlüsselt, in Buchstaben umgewandelt und ausgegeben.

Befindet sich ein errechneter ASCII-Code jedoch nicht im vorgesehenen Bereich (0 bis 255), sprich: waren die eingegebenen Codezahlen falsch, wird das Programm mit entsprechender Fehlermeldung abgebrochen (Zeile 80).

Es kann auch vorkommen, daß Sie einen unleserlichen Zeichensalat auf den Bildschirm bekommen. Auch dann waren die Codes falsch, jedoch hielt sich die Abweichung so sehr in Grenzen, daß es rechnerisch nicht einwandfrei überprüfbar war. Beachten Sie auch, daß Sie für jeden Tag nur einen Eintrag in Ihr »Tagebuch« vornehmen dürfen.

Benutzer anderer Computer als dem C 64 müssen den POKE gegen RUN/STOP-RESTORE (in den Zeilen 10 und 99) gegebenenfalls ändern oder ganz weglassen.

(Peter T. Schmidt/sk)

Sprites suchen

```
1 V=53248:POKEV+1,128:POKEV,128:POKEV+23,255:
  POKEV+29,255:POKEV+21,255
2 PRINT"{CLR}";A
3 GETA$:IFA$=""THEN3
4 IFA$="{CRSR-Rechts}"
  THENA=A+1:IFA=256THENA=255
5 IFA$="{CRSR-nach unten}"
  THENA=A-1:IFA=-1 THENA=0
6 IFA$="M"THENPOKEV+28,255
7 IFA$="N"THENPOKEV+28,0
8 POKE2040,A:GOTO2
```

Das obige Programm »Sprite-View« muß abgetippt und gespeichert werden. Nach dem Start erscheint ein weißes Sprite auf dem Bildschirm und in der linken oberen Ecke steht eine Zahl. Dies ist die Blockzahl des Sprites. Multipliziert man die Blockzahl mit 64, so erhält man die Startadresse des Sprites. Mit der Taste <Cursor-rechts> erhöht man die Blockzahl um 1, mit <Cursor-hinunter> er-

niedrigt man sie um 1. Mit <M> schaltet man den Multicolormodus des Sprites ein, mit <N> wieder aus. So kann man zwischen 256 Sprites blättern.

Programm-Erläuterung:

Zeile 1: Initialisierung Sprite 0
Zeile 2: druckt Blockzahl auf Bildschirm
Zeile 3: warte auf Taste
Zeile 4: wenn Taste = Cursor rechts, dann Blockzahl +1, wenn Blockzahl = 256, dann Blockzahl = 255
Zeile 5: wenn Taste = Cursor runter, dann Blockzahl -1, wenn Blockzahl = -1, dann Blockzahl 0
Zeile 6: wenn Taste = M, dann Multicolor ein
Zeile 7: Wenn Taste = N, dann Multicolor aus
Zeile 8: Blockzahl POKEn, Sprung nach 2

(Christoph Brochhaus/sk)

Was tun bei »out of memory«?

Hatten Sie auch schon mal während der Arbeit mit dem C 64 plötzlich aus heiterem Himmel einen »out of memory error«? Abgesehen davon, daß Sie wirklich ein äußerst langes Programm im Speicher haben könnten, taucht diese Meldung auch nach dem Laden von vielen Maschinenprogrammen auf. Die Fehlermeldung kommt meistens dann, wenn Sie nach dem Laden entweder ein Basic-Programm mit RUN starten, oder aber ein weiteres Programm laden möchten.

Warum taucht dieser Fehler auf? Wenn Sie zum Beispiel eine Variable mit einem Wert belegen (A=10), dann legt der C 64 diese Zahl 10 direkt hinter dem aktuellen (Basic-)Programm im Speicher ab. Dazu muß er aber wissen, wo das Programm aufhört. Zu diesem Zweck gibt es einen sogenannten »Zeiger«, der die letzte vom Programm belegte Adresse angibt. Der Zeiger wird aktualisiert, wenn Sie das Programm in seiner Länge verändern, also eine Zeile hinzuschreiben oder löschen. Der Zeiger wird aber auch gesetzt, wenn Sie ein Programm von Diskette laden.

Maschinenprogramme werden vom Programmierer meistens in Speicherbereiche gelegt, die von Basic-Programmen aus normalerweise nicht überschrieben werden. Diese Bereiche haben sehr hohe Adressen, auf die der eben beschriebene Programmende-Zeiger nach dem Laden gesetzt wird. Wenn Sie jetzt einen Befehl eingeben, der eine Variable definiert, erkennt der C64, daß hinter dem Programmende-Zeiger bereits ein Speicherbereich liegt, der von einem Basic-Programm nicht benutzt werden darf.

Ein Beispiel: Basic-Programme haben im allgemeinen den Speicherbereich von Adresse 2049 bis Adresse 40959 zur Verfügung. Maschinenprogramme verwenden jedoch häufig den etwas höher liegenden Bereich zwischen Adresse 49152 und Adresse 53247. Wenn ein Programm in diesen hohen Bereich geladen, also der Zeiger zum Beispiel auf Adresse 51234 gesetzt wird, kommt es beim späteren Belegen einer Variablen zum »out of memory error«.

Auch beim Laden von Programmen wird eine Variable belegt: Der C64 merkt sich den Namen des zu ladenden Programms am Ende des Basic-Speichers. Vorher wird jedoch anhand des Programmende-Zeigers überprüft, ob dafür überhaupt noch Platz ist. In unserem Beispiel mit der Adresse 51234 ist dies nicht der Fall. Also kommt es auch beim Nachladen von Programmen zum »out of memory error«.

Doch was tun? Geben Sie einfach NEW ein. Keine Angst: Dadurch wird das Maschinenprogramm nicht gelöscht. Lediglich der Programmende-Zeiger wird auf Adresse 2051 gesetzt, also festgelegt, daß sich kein Basic-Programm mehr im Speicher befindet. (Michael Thomas/sk)

Sicher ist sicher

Das kleine Programm »TIMESAVE« ist für den Basic-Programmierer eine unerläßliche Hilfe. Es speichert Ihr Programm nach Ablauf einer gewissen Zeit selbständig auf Diskette.

Eine goldene Regel beim Programmieren und Abtippen heißt: Programm vor dem Start speichern. Jedem ist es schon einmal passiert: Nach RUN war der C64 im Nirwana verschwunden und nur durch Aus-/Einschalten oder Reset wieder auf den Boden der Tatsachen zurückzuholen. Dann heißt es: Alles noch mal eingeben. Der Sicherheitskopierer »TIMESAVE« nimmt dem Programmierer diese Sorge ab.

Denn dieses Programm speichert in frei wählbaren, regelmäßigen Abständen das Basic-Programm – sogar aus laufenden Basic-Programmen heraus – auf Diskette. Da das Programm IRQ und den NMI-Vektor benutzt, wird aber nicht aus laufenden Maschinenprogrammen gespeichert, die diese Vektoren verändern.

```
Name : timesave              c000 c1bc
--------------------------------------
c000 : 20 75 c1 78 a9 05 8d 14   7b
c008 : 03 a9 c1 8d 15 03 58 a9   20
c010 : 7f 8d 0d dd a9 5d 8d 18   41
c018 : 03 a9 c0 8d 19 03 a9 00   22
c020 : 85 fc 85 fd 85 fe a9 40   bc
c028 : 8d 74 c1 ad 0d dd a2 00   60
c030 : bd 3e c0 c9 00 f0 25 20   d2
c038 : d2 ff e8 4c 30 c0 0d 12   2f
c040 : 54 49 4d 45 53 41 56 45   58
c048 : 20 49 53 54 20 49 4e 53   98
c050 : 54 41 4c 4c 49 45 52 54   92
c058 : 2e 92 0d 00 60 48 8a 48   16
c060 : 98 48 a9 7f 8d 0d dd ad   8b
c068 : 0d dd 10 03 4c 72 fe ad   78
c070 : 8d 02 c9 04 f0 3f 20 e1   3e
c078 : ff d0 2b 20 15 fd 20 a3   b7
c080 : fd 20 18 e5 a9 5d 8d 18   3c
c088 : 03 a9 c0 8d 19 03 ad 0d   bc
c090 : dd 78 a9 05 8d 14 03 a9   8d
c098 : c1 8d 15 03 58 68 68 68   01
c0a0 : 68 68 68 6c 02 a0 20 42   0e
c0a8 : c1 ad 0d dd 68 68 68 68   7b
c0b0 : 68 68 6c 02 a0 a9 7f 8d   18
c0b8 : 0d dd ad 0d dd a9 47 8d   24
c0c0 : 18 03 a9 fe 8d 19 03 78   43
c0c8 : a9 31 8d 14 03 a9 ea 8d   34
c0d0 : 15 03 58 a2 00 bd e3 c0   d0
c0d8 : c9 00 f0 26 20 d2 ff e8   0d
c0e0 : 4c d5 c0 0d 12 54 49 4d   6c
c0e8 : 45 53 41 56 45 20 49 53   13
c0f0 : 54 20 41 55 53 47 45 53   7b
c0f8 : 43 48 41 4c 54 45 54 92   1f
c100 : 0d 00 4c bc fe a5 fc c9   5d
c108 : 3b f0 05 e6 fc 4c 31 ea   a6
c110 : a9 00 85 fc a5 fd c9 3b   a2
c118 : f0 05 e6 fd 4c 31 ea a9   51
c120 : 00 85 fd a5 fe c9 0f b0   f3
c128 : 05 e6 fe 4c 31 ea a9 00   fb
c130 : 85 fc 85 fd 85 fe 20 42   aa
c138 : c1 68 68 68 68 68 68 6c   99
c140 : 02 a0 ee 74 c1 a2 08 20   6e
c148 : ba ff a9 10 a2 65 a0 c1   ca
c150 : 20 bd ff a9 01 85 b0 a9   d6
c158 : 08 85 b1 a9 b0 a6 2d a4   03
c160 : 2e 20 d8 ff 60 00 00 00   da
c168 : 00 00 00 00 00 00 00 00   69
c170 : 00 00 00 00 00 a2 00 bd   01
c178 : 85 c1 c9 00 f0 15 20 d2   2e
c180 : ff e8 4c 77 c1 0d 12 46   4f
c188 : 49 4c 45 4e 41 4d 45 20   e6
c190 : 3e 92 00 a9 00 85 c6 a9   e7
c198 : 14 20 d2 ff a2 00 a9 00   42
c1a0 : 85 c6 a5 c6 f0 fc ad 77   67
c1a8 : 02 9d 65 c1 20 d2 ff e8   75
c1b0 : e0 0f d0 ea a9 0d 20 d2   d2
c1b8 : ff 60 00 ff 00 ff 00 ff   e7
```

Listing 1. »TIMESAVE« Bitte mit MSE (Seite 159) eingeben

»TIMESAVE« (Listing 1) steht bei $C000 im Speicher und belegt somit keinen Basic-Speicherplatz. Nach dem Laden mit

```
LOAD "TIMESAVE",8,1
```

muß NEW eingegeben werden. Mit SYS 49152 startet man dann das Programm. Zuerst gibt man den Filenamen ein, unter dem die Sicherheitskopien des Basic-Speichers auf Diskette abgelegt werden sollen. Dabei ist zu beachten, daß der Filename 15 Zeichen lang sein muß. Es werden dabei auch Leerzeichen in Filenamen akzeptiert. Die ersten Zeichen sollten jedoch Buchstaben oder Zahlen sein. Nach dieser Eingabe kommt die Meldung:

»TIMESAVE IST INSTALLIERT«

Variabler Zeitabstand

Nun stellt man den die gewünschte Zeit zwischen zwei Sicherheitskopien ein. Dies geschieht mit POKE 49446,X (0 ⟨ X ⟨ =255). Die Variable X gibt den Zeitabstand in Minuten an, ändert man diese Speicherstelle nicht, so speichert »TIMESAVE« das Programm alle 15 Minuten.

Speichern im 15-Minuten-Takt

Wenn Sie außerhalb dieser Zeiten Ihr Programm speichern wollen, steht Ihnen dazu die RESTORE-Taste zur Verfügung. Mit der Tastenkombination ⟨CTRL RESTORE⟩ klinkt man die Routine aus dem Interrupt-System aus. Ein Neustart erfolgt mit SYS 49152.

(Thomas Olschewski/U. Beenen)

Kurzübersicht	
(Neu)-Start	SYS49152
X-Minutenabstand	POKE 49446,X
Sofort Speichern	⟨RESTORE⟩
Ausklinken	⟨CTRL RESTORE⟩
Filename	Immer 15 Zeichen

Für alle Programmierer: vier Utilities mit Pfiff

Jetzt können Sie Programme vergleichen und erfahren dabei genau, wie sich diese unterscheiden. Mit inversen REM-Zeilen lassen sich Basic-Programme noch besser strukturieren. Strings werden extrem schnell sortiert und zu guter Letzt haben Sie 40 Byte des Speichers ständig im Blick.

Wer achtet nicht bei der Entwicklung von eigenen Programmen auf eine gute Struktur? Schließlich will man auch nach einem Monat noch den Überblick behalten. Ein wesentliches Element dabei ist die Dokumentation mit REM-Zeilen.

Übersicht durch inverse REM-Zeilen

REM-Zeilen machen Listings übersichtlicher. Ist jedoch der Kommentartext zusätzlich invertiert, erleichtert dies das Auffinden bestimmter Programmteile.

Mit dem Programm »Invers REM« (Listing 1) werden sämtliche Kommentartexte bei einem REM-Befehl invertiert, das heißt »revers« dargestellt. Dazu wird das erste Zeichen nach dem REM-Befehl mit dem Revers-Zeichen (CHR$-Code 18) überschrieben. Nach dem REM-Befehl sollte also mindestens ein Leerzeichen stehen, damit kein Informationstext verlorengeht.

Sind die REM-Zeilen einmal konvertiert worden, kann man das Basic-Programm abspeichern. Die Kommentarzeilen werden dann immer hervorgehoben dargestellt.

Geben Sie Listing 1 mit dem MSE ein. »Invers REM« wird absolut geladen, das heißt mit »LOAD "Name",8,1«. Danach muß »NEW« eingegeben werden. Sie können nun Ihr eigenes Basic-Programm eingeben oder von Diskette laden. Nach »SYS 49152« und ⟨RETURN⟩ werden alle REM-Zeilen konvertiert und damit invers. Jetzt wird das so geänderte Programm gespeichert.

Noch eines sei gesagt. Da modifizierte Betriebssysteme, wie zum Beispiel »Speeddos Plus«, eine veränderte List-Routine besitzen, werden die REM-Kommentare nicht invertiert ausgegeben.

Speicher-Spion

Das Programm »Showmen.irq« (Listing 2) zeigt Ihnen ständig einen beliebigen Speicherbereich mit 40 Byte Länge in der ersten Bildschirmzeile (Bild 1).

Dieses interruptgesteuerte Programm eignet sich hervorragend zur Fehlersuche, da man sehr schnell feststellen kann, ob und wie sich ein Speicherbereich ändert. Dazu werden in der ersten Bildschirmzeile die Bildschirmcodes des ausgewählten Speicherbereichs dargestellt. Sie können jetzt nur noch auf die restlichen 24 Zeilen des Bildschirms zugreifen.

```
SYS49152
ADRESSE: C000
READY.
```

Bild 1. Die Kopfzeile unseres Programms »Showmen.irq« mit Einblick in den Speicher. Hier wurden die 40 Byte ab Adresse $C000 gewählt. Durch die ständige Kontrolle lassen sich Änderungen schnell erkennen.

Geben Sie Listing 2 mit dem MSE ein. Geladen wird es absolut mit »LOAD "Showmem.irq",8,1«. Anschließend muß noch »NEW« eingegeben werden.

Sie starten diesen Speicher-Spion mit »SYS 49152«. Das Programm gibt dann einige Kommentarzeilen aus und fordert zur Eingabe der Anfangsadresse auf. Ab dieser

Adresse werden die folgenden 40 Byte auf dem Bildschirm dargestellt. Die Eingabe der Adresse muß hexadezimal in vierstelliger Form erfolgen. Wird dieses Format nicht eingehalten, muß man die Eingaben wiederholen. Anschließend befindet man sich wieder im Direktmodus und kann in der ersten Bildschirmzeile die ausgewählten Speicherinhalte betrachten. Mit ⟨RUN/STOP RESTORE⟩ wird die Routine beendet.

Extrem schnelles String-Sortieren

String-Sortieren ist eine komplizierte und langsame Arbeit für den C64, aber nicht mit »Sort String«. Diese Routine (Listing 3) hat die hervorragende Sortiergeschwindigkeit von 100 Strings in zirka 1 Sekunde und 1000 Strings in zirka 25 Sekunden.

Eine solche Geschwindigkeit kann nur durch ein Maschinenprogramm erreicht werden. Der Trick liegt darin, daß nur die Element-Pointer vertauscht werden. Die Strings selbst bleiben jedoch unberührt im Speicher. Dadurch entfällt auch die zeitaufwendige »Garbage Collection«.

Das Maschinenprogramm ist als Basic-Lader, den man mit »GOSUB 50000« aufruft, programmiert. Zuvor muß aber noch die Speichergrenze auf $9d9f (dez. 40351) herabgesetzt und der CLR-Befehl ausgeführt werden. Dies alles geschieht durch:

```
1 POKE 56,157 : POKE 55,159 : CLR : GOSUB 50000
```

Nun können Sie Ihr String-Array dimensionieren und bearbeiten. Beachten Sie aber dabei, daß nur eindimensionale String-Arrays sortiert werden. Bevor Sie die Sortierroutine aufrufen, müssen einige Vorbereitungen erfolgen.

Dem Programm muß der Variablenname mitgeteilt werden:

```
POKE 40352,ASC (1.Buchstabe des Variablennamen)
POKE 40353,ASC(2.Buchstabe des Variablennamen)
```

Besteht der Variablennamen nur aus einem Buchstaben, dann wird in die zweite Speicherstelle der Wert 128 geschrieben:

```
POKE 40353,128
```

Nun muß dem Programm die Elementanzahl mitgeteilt werden. Das High-Byte wird in der Adresse 40355 und das Low-Byte wird in der Adresse 40354 abgelegt:

```
POKE 40354,Elementanzahl AND 255
POKE 40355,INT(Elementanzahl/256)
```

Die Maschinenroutine wird nun durch SYS 40371 aufgerufen. Danach sollte der Inhalt der Speicherstelle 40356 ausgelesen werden, denn dieser Wert zeigt einen eventuellen Fehler an. Bei dem Wert 1 wurde das Array nicht gefunden. Eine 2 zeigt an, daß das Array nicht eindimensional ist. Jeder andere Wert gibt zu erkennen, daß das String-Array sortiert wurde.

Geben Sie Listing 3 mit dem Checksummer ein. Listing 4 ist ein Demoprogramm, das 20 zufällige Strings mit dieser Routine sortiert. Es kann entweder in Listing 3 anstelle der Zeile 100 eingefügt werden, oder Sie laden und starten Listing 4, nachdem die Sortierroutiene von Listing 3 generiert wurde. Beachtenswert ist bei dem Demo, daß der Sortiervorgang erst nach dem Tastendruck (Zeile 1045) eingeleitet wird. Lassen Sie sich überraschen von der Sortiergeschwindigkeit.

Programme vergleichen mit Komfort

Haben Sie sich auch schon darüber geärgert, daß man mit dem VALIDATE-Befehl zwar erfährt, ob zwei Programme gleich sind oder nicht; Sie erfahren aber nicht, wo sich diese überall unterscheiden und wie. Das wird jetzt anders.

Das Programm »Compare Files« (Listing 5) vergleicht 2 Files auf der Diskette. Zuerst fordert das Programm zur Eingabe des ersten Filenamen auf. Es wird nun versucht, das File auf der Diskette zu finden und zu laden. Findet das Programm dieses File nicht, so gibt es eine Meldung auf den Bildschirm aus und fordert eine neue Eingabe. Dies wird so lange wiederholt, bis das erste File geladen wurde. Jetzt muß man den zweiten Filenamen eingeben; das geschieht wie beim ersten.

Nach dem Laden vergleicht das Programm die beiden Dateien. Nicht übereinstimmende Bytes werden mit »Offset« angezeigt. Ein Beispiel:

Offset: $004d File 1: $41 a File 2: $00 .

In diesem Beispiel hat das 79. Byte ($4d) des ersten Files den Inhalt $41, das 79. Byte im zweiten File dagegen den Inhalt $00. »a« und ».« sind die ASCII-Zeichen der Werte $41 und $00. Eine Ausgabe von »$..« bedeutet, daß dieses Byte bei diesem File nicht mehr vorhanden ist (dieses File ist also kürzer).

Die Ausgaben der Offset-Tabelle können Sie mit ⟨CTRL⟩ verlangsamen. ⟨RUN/STOP⟩ hält die Ausgabe an, mit ⟨SPACE⟩ setzt man die Ausgaben wieder fort. Am Ende des Vergleichs wird die Anzahl der unterschiedlichen Bytes ausgegeben.

Wenn der Vergleich abgeschlossen ist, wird man vor die Wahl gestellt, ob der Vergleich wiederholt werden soll. Man antwortet mit den Tasten ⟨J⟩ für ja oder ⟨N⟩ für nein. Bei nein wird das Programm beendet.

Geben Sie Listing 5 mit dem MSE ein. »Compare Files« lädt man absolut. Vor dem Start müssen Sie noch »NEW« eingeben. Mit der Eingabe von »SYS 49152« startet man das Programm.

Das Programm eignet sich beispielsweise auch hervorragend zum Auffinden der Unterschiede zwischen verschiedenen Kernel-Versionen, wenn diese softwaremäßig vorliegen.

(Andreas Nagler/Thomas Lipp)

Listing 1. »Invers REM« invertiert sämtliche Kommentartexte nach dem REM-Befehl. Bitte mit dem MSE (Seite 159) eingeben.

```
Name : invers rem              c000 c056
-----------------------------------------
c000 : a2 03 a0 05 20 42 c0 d0   a5
c008 : 1b ca f0 42 20 46 c0 20   84
c010 : 42 c0 f0 f5 8a 18 69 05   c6
c018 : aa 20 46 c0 ca e0 04 b0   a1
c020 : f8 20 42 c0 c9 8f f0 06   ba
c028 : 20 46 c0 4c 04 c0 20 46   78
c030 : c0 ad 44 c0 8d 3b c0 a9   f9
c038 : 12 99 00 0b 20 46 c0 4c   48
c040 : 04 c0 b9 00 08 60 c8 f0   9b
c048 : 01 60 ee 44 c0 60 a9 08   83
c050 : 8d 44 c0 60 60 60 ff 00   44
```

Listing 2. »Showmem.irq« zeigt ständig einen beliebigen Speicherbereich von 40 Byte Länge in einer Kopfzeile. Bitte mit dem MSE (Seite 159) eingeben.

```
Name : showmem.irq             c000 c099
-----------------------------------------
c000 : 78 a9 31 8d 14 03 a9 ea   21
c008 : 8d 15 03 58 a9 39 a0 c0   54
c010 : 20 1e ab 20 52 c0 8e 2d   ee
c018 : c0 8d 2c c0 78 a9 29 8d   57
c020 : 14 03 a9 c0 8d 15 03 58   76
c028 : 60 a0 00 b9 00 10 99 00   f7
c030 : 04 c8 c0 27 d0 f5 4c 31   fe
c038 : ea 0d 41 44 52 45 53 53   c5
c040 : 45 3a 20 00 20 cf ff c9   bf
c048 : 0d d0 06 4c 44 c0 20 cf   33
c050 : ff 60 20 5a c0 aa 20 5a   69
c058 : c0 60 a9 00 8d 90 c0 20   53
c060 : 44 c0 c9 20 d0 0b 20 44   e9
c068 : c0 4c 62 c0 c9 20 d0 0e   fc
c070 : 60 20 85 c0 0a 0a 0a 0a   87
c078 : 8d 90 c0 20 44 c0 20 85   57
c080 : c0 0d 90 c0 60 c9 3a 08   50
c088 : 29 0f 28 90 02 69 08 60   a1
c090 : 00 16 c1 bd 88 c7 20 16   37
c098 : c1 ff 00 ff 00 ff 00 ff   59
```

```
1 POKE 56,157:POKE 55,159:CLR:GOSUB 50000   <104>
100 END                                      <102>
49990 REM******************************      <003>
50000 FOR F=40371 TO F+429:READ A:POKE F,A
      :NEXT                                  <100>
50008 RETURN                                 <025>
50009 DATA 173,160,157,41,127                <113>
50010 DATA 141,160,157,173,161               <142>
50020 DATA 157,9,128,141,161                 <245>
50030 DATA 157,169,0,141,164                 <213>
50040 DATA 157,165,47,133,180                <083>
50050 DATA 165,48,133,181,160                <220>
50060 DATA 0,177,180,205,160                 <118>
50070 DATA 157,208,8,200,177                 <113>
50080 DATA 180,205,161,157,240               <151>
50090 DATA 42,160,2,177,180                  <221>
50100 DATA 141,165,157,200,177               <054>
50110 DATA 180,141,166,157,24                <202>
50120 DATA 165,180,109,165,157               <066>
50130 DATA 133,180,165,181,109               <166>
50140 DATA 166,157,133,181,197               <042>
50150 DATA 50,144,207,240,205                <169>
50160 DATA 169,1,141,164,157                 <204>
50170 DATA 76,224,158,160,4                  <149>
50180 DATA 177,180,201,1,240                 <166>
50190 DATA 8,169,2,141,164                   <234>
50200 DATA 157,76,224,158,24                 <165>
50210 DATA 165,180,105,7,133                 <001>
50220 DATA 180,165,181,105,0                 <104>
50230 DATA 133,181,173,162,157               <237>
50240 DATA 141,177,157,173,163               <197>
50250 DATA 157,141,178,157,173               <231>
50260 DATA 178,157,208,12,173                <092>
50270 DATA 177,157,240,4,201                 <165>
50280 DATA 1,208,3,76,224                    <116>
50290 DATA 158,78,178,157,110                <108>
50300 DATA 177,157,56,173,162                <134>
50310 DATA 157,237,177,157,141               <030>
50320 DATA 175,157,173,163,157               <210>
50330 DATA 237,178,157,141,176               <066>
50340 DATA 157,162,0,138,141                 <034>
50350 DATA 168,157,141,169,157               <221>
50360 DATA 173,177,157,141,170               <086>
50370 DATA 157,173,178,157,141               <246>
50380 DATA 171,157,238,168,157               <001>
50390 DATA 208,3,238,169,157                 <245>
50400 DATA 173,169,157,205,176               <198>
50410 DATA 157,240,4,176,85                  <140>
50420 DATA 144,10,173,168,157                <020>
50430 DATA 205,175,157,240,2                 <126>
50440 DATA 176,73,238,170,157                <076>
50450 DATA 208,3,238,171,157                 <111>
50460 DATA 160,3,165,180,133                 <091>
50470 DATA 184,133,186,165,181               <052>
50480 DATA 133,185,133,187,24                <207>
50490 DATA 165,184,109,168,157               <198>
50500 DATA 133,184,165,185,109               <044>
50510 DATA 169,157,133,185,24                <219>
50520 DATA 165,186,109,170,157               <217>
50530 DATA 133,186,165,187,109               <082>
50540 DATA 171,157,133,187,136               <186>
50550 DATA 208,223,32,225,158                <121>
50560 DATA 173,167,157,240,163               <238>
50570 DATA 48,161,32,80,159                  <131>
50580 DATA 162,1,76,115,158                  <236>
50590 DATA 138,208,129,76,52                 <218>
50600 DATA 158,96,160,0,140                  <209>
50610 DATA 167,157,177,184,141               <218>
50620 DATA 172,157,177,186,141               <088>
50630 DATA 173,157,200,152,205               <119>
50640 DATA 172,157,240,2,176                 <186>
50650 DATA 15,205,173,157,240                <027>
50660 DATA 25,144,23,169,1                   <216>
50670 DATA 141,167,157,76,79                 <011>
50680 DATA 159,205,173,157,240               <052>
50690 DATA 2,176,64,169,255                  <116>
50700 DATA 141,167,157,76,79                 <043>
50710 DATA 159,140,165,157,160               <021>
50720 DATA 1,177,184,133,182                 <100>
50730 DATA 200,177,184,133,183               <071>
50740 DATA 172,165,157,136,177               <154>
50750 DATA 182,141,174,157,140               <094>
50760 DATA 165,157,160,1,177                 <035>
50770 DATA 186,133,182,200,177               <067>
50780 DATA 186,133,183,172,165               <167>
50790 DATA 157,177,182,200,205               <171>
50800 DATA 174,157,208,3,76                  <052>
50810 DATA 240,158,144,180,76                <064>
50820 DATA 15,159,96,160,2                   <169>
50830 DATA 177,184,72,177,186                <189>
50840 DATA 145,184,104,145,186               <109>
50850 DATA 136,16,243,96,0                   <006>
```

Listing 3. »Sort String« sortiert Strings extrem schnell. Bitte mit dem Checksummer (Seite 159) eingeben.

```
1000 N=20:DIM A$(N):PRINT"{CLR,2DOWN}UNSOR
     TIERTE STRINGS WERDEN GENERIERT"          <174>
1010 FOR I=1 TO N:A$="":FOR J=1 TO  20:B$=
     CHR$(INT(RND(0)*26+65)):A$=A$+B$         <002>
1020 NEXT:A$(I)=A$:NEXT                        <202>
1030 PRINT"{CLR}UNSORTIERT:":PRINT"ŪŪŪŪŪŪŪ
     ŪŪŪ":FOR I=1 TO N:PRINT A$(I):NEXT       <017>
1040 PRINT"{DOWN}SORTIEREN: TASTE DRUECKEN
     "                                         <209>
1045 GET A$:IF A$=""THEN 1045                  <213>
1050 POKE 40352,65:REM 1.BUCHST.D.VAR          <224>
1060 POKE 40353,128:REM 2.STELLE LEER          <068>
1070 POKE 40355,INT(N/256)    :REM N           <003>
1080 POKE 40354,N-INT(N/256)*256:REM N         <073>
1090 SYS 40371:REM 9DB3                        <028>
1100 FE=PEEK(40356) :REM   FEHLERFLAG          <079>
1110 IF FE=1 THEN PRINT"FEHLER - ARRAY NIC
     HT GEFUNDEN":END                          <239>
1120 IF FE=2 THEN PRINT"FEHLER - ARRAY NIC
     HT EINDIMENSIONAL"                        <194>
1140 PRINT"{CLR}SORTIERT:":PRINT"ŪŪŪŪŪŪŪŪ"
     :FOR I=1 TO N:PRINT A$(I):NEXT:END        <193>
1150 FOR I=1 TO N:PRINT A$(I):NEXT:END         <020>
```

Listing 4. »Demo Sort String«, bitte mit dem Checksummer (Seite 159) eingeben.

```
Name : compare files        c000 c571
-------------------------------------
c000 : a9 37 85 01 a9 00 8d 8a   ac
c008 : 02 4c a9 c0 48 8a 20 12   30
c010 : c0 68 48 4a 4a 4a 4a 20   c0
c018 : 2a c0 aa 68 29 0f 20 2a   3a
c020 : c0 48 8a 20 d2 ff 68 4c   12
c028 : d2 ff 18 69 f6 90 02 69   fc
c030 : 06 69 3a 60 a9 08 a2 08   fb
c038 : a0 00 20 ba ff ad 5b c0   94
c040 : ae 60 c0 ac 61 c0 20 bd   fc
c048 : ff a9 00 ae 5c c0 ac 5d   2b
c050 : c0 20 d5 ff 8e 5e c0 8c   8e
c058 : 5f c0 60 0b 01 08 03 00   ed
c060 : 62 c0 53 4f 52 54 20 53   d0
c068 : 54 52 49 4e 47 0d 00 00   de
c070 : 00 00 00 00 00 00 00 00   71
c078 : 00 00 00 00 00 00 00 00   79
c080 : 00 00 00 00 00 00 00 00   81
c088 : 00 00 00 00 00 00 00 00   89
c090 : 00 00 00 00 00 00 00 00   91
c098 : 00 00 00 00 00 01 08 00   c1
c0a0 : 00 00 00 00 00 00 00 00   a1
c0a8 : 00 a9 00 85 c6 20 90 ff   dd
c0b0 : 20 13 c3 a9 22 a0 c3 20   f6
c0b8 : 1e ab a0 00 20 cf ff 99   88
c0c0 : 62 c0 c8 c9 0d d0 f5 88   2e
c0c8 : 8c 5b c0 a9 62 8d 60 c0   fd
c0d0 : a9 c0 8d 61 c0 a9 01 8d   e1
c0d8 : 5c c0 8d 9d c0 a9 08 8d   40
c0e0 : 5d c0 8d 9e c0 20 34 c0   34
c0e8 : 90 0a a9 39 a0 c3 20 1e   f4
c0f0 : ab 4c b3 c0 ad 5e c0 8d   b2
c0f8 : 9f c0 ad 5f c0 8d a0 c0   cb
c100 : a9 67 a0 c3 20 1e ab ad   fa
c108 : 9d c0 ae 9e c0 20 0c c0   43
c110 : a9 78 a0 c3 20 1e ab ad   93
c118 : 9f c0 ae a0 c0 20 0c c0   96
c120 : a9 50 a0 c3 20 1e ab a0   75
c128 : 00 20 cf ff 99 7f c0 c8   56
c130 : c9 0d d0 f5 88 8c 5b c0   4e
c138 : a9 7f 8d 60 c0 a9 c0 8d   88
c140 : 61 c0 ad 9f c0 8d 5c c0   cc
c148 : ad a0 c0 8d 5d c0 20 34   ec
c150 : c0 90 0a a9 39 a0 c3 20   f8
c158 : 1e ab 4c 20 c1 ad 5e c0   e7
c160 : 8d a1 c0 ad 5f c0 8d a2   1b
c168 : c0 a9 67 a0 c3 20 1e ab   f8
```

Listing 5. »Compare Files«, vergleicht zwei Files und zeigt nicht übereinstimmende Bytes an. Bitte mit dem MSE (Seite 159) eingeben.

```
c170 : ad 9f c0 ae a0 c0 20 0c  9b
c178 : c0 a9 78 a0 c3 20 1e ab  4c
c180 : ad a1 c0 ae a2 c0 20 0c  cc
c188 : c0 a9 0d 20 d2 ff 20 d2  b7
c190 : ff 20 d2 ff a9 00 8d a7  74
c198 : c0 8d a8 c0 ad 9d c0 85  37
c1a0 : fc ad 9e c0 85 fd ad 9f  71
c1a8 : c0 85 fe ad a0 c0 85 ff  c6
c1b0 : a9 00 8d a5 c0 8d a6 c0  06
c1b8 : 8d a3 c0 8d a4 c0 a0 00  cb
c1c0 : b1 fc d1 fe d0 08 ad a3  8f
c1c8 : c0 0d a4 c0 f0 1d 20 81  cb
c1d0 : c2 ae a7 c0 e8 8e a7 c0  0e
c1d8 : d0 03 ee a8 c0 20 e1 ff  8f
c1e0 : d0 09 a9 00 85 c6 20 e4  78
c1e8 : ff f0 fb e6 fc a5 fc d0  cd
c1f0 : 02 e6 fd e6 fe a5 fe d0  7c
c1f8 : 02 e6 ff ee a5 c0 ad a5  ad
c200 : c0 d0 03 ee a6 c0 a5 fd  ca
c208 : cd a0 c0 d0 0a a5 fc cd  cd
c210 : 9f c0 d0 03 ce a3 c0 a5  fc
c218 : ff cd a2 c0 d0 0a a5 fe  b0
c220 : cd a1 c0 d0 03 ce a4 c0  c3
c228 : ad a3 c0 f0 91 ad a4 c0  90
c230 : f0 8c a9 34 a0 c5 20 1e  4c
c238 : ab ad a7 c0 ae a8 c0 20  2f
c240 : 0c c0 a9 20 20 d2 ff 20  f4
c248 : d2 ff ad a8 c0 ae a7 c0  3c
c250 : 20 cd bd a9 ea a0 c4 20  02
c258 : 1e ab a9 00 85 c6 20 e4  8f
c260 : ff f0 fb c9 4a f0 07 c9  eb
c268 : 4e f0 0e 4c 5e c2 a9 0d  f8
c270 : 20 d2 ff 20 d2 ff 4c 94  85
c278 : c1 a9 d4 a0 c4 4c 1e ab  d5
c280 : 60 a9 49 a0 c5 20 1e ab  48
c288 : ae a6 c0 ad a5 c0 20 0c  68
c290 : c0 a9 54 a0 c5 20 1e ab  7b
c298 : ad a3 c0 f0 0a a9 68 a0  36
c2a0 : c5 20 1e ab 4c ae c2 a0  f9
c2a8 : 00 b1 fc 20 12 c0 a9 20  d2
c2b0 : 20 d2 ff ad a3 c0 d0 0a  86
c2b8 : a0 00 b1 fc 20 fe c2 4c  02
c2c0 : c7 c2 a9 2e 20 d2 ff a9  04
c2c8 : 5e a0 c5 20 1e ab ad a4  2b
c2d0 : c0 f0 0a a9 68 a0 c5 20  a3
c2d8 : 1e ab 4c e4 c2 a0 00 b1  10
c2e0 : fe 20 12 c0 a9 20 20 d2  4d
c2e8 : ff ad a4 c0 d0 0a a0 00  df
c2f0 : b1 fe 20 fe c2 4c fd c2  14
c2f8 : a9 2e 20 d2 ff 60 48 29  91
c300 : 7f c9 20 90 08 c9 7f b0  ac
c308 : 04 68 4c d2 ff 68 a9 2e  f4
c310 : 4c d2 ff a9 7f a0 c3 20  47
c318 : 1e ab a9 77 a0 c4 20 1e  52
c320 : ab 60 0d 66 49 4c 45 4e  b4
c328 : 41 4d 45 20 44 45 53 20  61
c330 : 31 2e 66 49 4c 45 53 3a  ec
c338 : 00 0d 91 66 49 4c 45 20  3c
c340 : 4e 49 43 48 54 20 47 45  fb
c348 : 46 55 4e 44 45 4e 2e 00  d4
c350 : 0d 66 49 4c 45 4e 41 4d  d3
c358 : 45 20 44 45 53 20 32 2e  c2
c360 : 66 49 4c 45 53 3a 00 0d  48
c368 : 66 49 4c 45 20 47 45 48  10
c370 : 54 20 56 4f 4e 20 24 00  ca
c378 : 20 42 49 53 20 24 00 93  c0
c380 : 64 49 45 53 45 53 20 70  95
c388 : 52 4f 47 52 41 4d 4d 20  92
c390 : 56 45 52 47 4c 45 49 43  a1
c398 : 48 54 20 32 20 70 72 67  77
c3a0 : 2d 66 49 4c 45 53 20 20  8c
c3a8 : 4d 49 54 45 49 4e 41 4e  00
c3b0 : 44 45 52 2e 0d 62 45 49  7d
c3b8 : 20 61 42 57 45 49 43 48  41
c3c0 : 55 4e 47 45 4e 20 57 45  85
c3c8 : 52 44 45 4e 20 44 49 45  2b
c3d0 : 20 6f 46 46 53 45 54 0d  cd
c3d8 : 61 44 52 45 53 53 45 20  be
c3e0 : 5a 55 4d 20 70 52 4f 47  a2
c3e8 : 52 41 4d 4d 53 54 41 52  59
c3f0 : 54 20 53 4f 57 49 45 20  28
c3f8 : 44 49 45 0d 42 45 49 44  d0
c400 : 45 4e 20 69 4e 48 41 4c  66
c408 : 54 45 20 41 55 53 47 45  c7
c410 : 47 45 42 45 4e 2e 0d 6d  98
c418 : 49 54 20 44 45 52 20 73  6a
c420 : 74 6f 70 2d 74 41 53 54  55
c428 : 45 20 4b 41 4e 4e 20 44  d9
c430 : 45 52 20 61 55 53 44 52  78
c438 : 55 43 4b 0d 41 4e 47 45  d2
c440 : 48 41 4c 54 45 4e 20 57  bc
c448 : 45 52 44 45 4e 2e 0d 62  bf
c450 : 45 49 4d 20 57 45 49 54  ff
c458 : 45 52 45 52 20 64 52 55  7b
c460 : 43 4b 20 41 55 46 20 44  0a
c468 : 49 45 20 73 74 6f 70 2d  a9
c470 : 74 41 53 54 45 0d 00 53  48
c478 : 54 4f 50 50 54 20 44 41  6c
c480 : 53 20 70 52 4f 47 52 41  45
c488 : 4d 4d 20 42 45 49 20 44  74
c490 : 45 52 20 4e 41 45 43 48  ac
c498 : 53 54 45 4e 0d 61 55 53  08
c4a0 : 47 41 42 45 2e 20 6a 45  d9
c4a8 : 44 45 20 41 4e 44 45 52  80
c4b0 : 45 20 74 41 53 54 45 20  78
c4b8 : 53 45 54 5a 54 20 44 49  f8
c4c0 : 45 20 20 20 20 61 55 53  2b
c4c8 : 47 41 42 45 20 46 4f 52  ff
c4d0 : 54 2e 0d 00 0d 0d 65 4e  ea
c4d8 : 44 45 20 44 45 53 20 70  a0
c4e0 : 52 4f 47 52 41 4d 4d 45  34
c4e8 : 53 00 0d 0d 73 4f 4c 4c  9c
c4f0 : 20 44 45 52 20 76 45 52  3d
c4f8 : 47 4c 45 49 43 48 20 44  5f
c500 : 49 45 53 45 52 20 42 45  23
c508 : 49 44 45 4e 20 66 49 4c  81
c510 : 45 53 20 20 4e 4f 43 48  08
c518 : 45 49 4e 4d 41 4c 20 47  c5
c520 : 45 4d 41 43 48 54 20 57  1b
c528 : 45 52 44 45 4e 20 28 4a  6b
c530 : 2f 4e 29 00 0d 61 4e 5a  9b
c538 : 41 48 4c 20 44 45 52 20  ac
c540 : 66 45 48 4c 45 52 3a 20  f5
c548 : 00 0d 4f 46 46 53 45 54  28
c550 : 3a 20 24 00 20 46 49 4c  95
c558 : 45 31 3a 20 24 00 20 46  18
c560 : 49 4c 45 32 3a 20 24 00  9c
c568 : 2e 2e 00 14 15 16 17 18  bf
c570 : 00 00 ff 00 ff 00 ff 00  70
```

Listing 5. (Schluß)

Die Speicherzellen 0 bis 1023, nach Funktionen geordnet

Dezimal	Hex	Funktion
Bandoperationen		
146	$92	Zeitkonstante beim Lesen vom Band
147	$93	Flagge für LOAD oder VERIFY
150	$96	Arbeitsspeicher für Band-Leseroutinen
153	$99	Nummer des Eingabegerätes
155	$9B	Fehlerkontrolle bei Bandoperationen
156	$9C	Flagge für korrektes Byte vom Band
158–159	$9E–$9F	Zwischenspeicher bei Kassettenoperationen
165	$A5	Zähler für Band-Synchronisierung
167	$A7	Zwischenspeicher für Kassettenroutinen
168	$A8	Bitzähler bei Band-Ein-/Ausgabe
170	$AA	Zwischenspeicher für Kassettenroutinen
171	$AB	Quersummenprüfung und Zähler für Band-Header
172–173	$AC-$AD	Zeiger auf die Anfangsadresse für Ein-/Ausgabe
174–175	$AE–$AF	Zeiger auf die Endadresse für Ein-/Ausgabe
176–177	$B0–$B1	Zeitkonstante beim Lesen vom Band
178–179	$B2–$B3	Zeiger auf den Kassettenpuffer
181	$B5	Blockangabe bei Kassettenoperationen
182	$B6	Ausgabe-Zwischenspeicher
183	$B7	Länge des File-Namens
185	$B9	Sekundär-Adresse
186	$BA	Geräte-Nummer
187–188	$BB–$BC	Zeiger auf Adresse des derzeitigen File-Namens
189	$BD	Zwischenspeicher für Zeichen
190	$BE	Blockzähler für Kassetten-Ein-/Ausgabe
191	$BF	Zwischenspeicher für LOAD-Operationen vom Band
192	$CO	Sperre des Motors der Datasette
193–194	$C1–$C2	Anfangsadresse für Ein-/Ausgabe-Operationen
195–196	$C3–$C4	Zeiger auf den Anfang des Programms hinter dem Tape Header
256–318	$100-$13E	Arbeitsspeicher für Fehler bei der Eingabe vom Band
Bildschirm-Cursor		
9	$9	Spaltenposition des Cursors vor dem letzten TAB- oder SPC-Befehl
200	$C8	Zeiger auf das Ende der eingegebenen logischen Zeile
201-202	$C9–$CA	Zeiger auf Zeilen- und Spaltenposition des letzten Zeichens einer Zeile
204	$CC	Schalter für Cursorblinken
205	$CD	Zähler für Blinkfrequenz des Cursors
206	$CE	Bildschirmcode des Zeichens unter dem Cursor
207	$CF	Flagge für Blinkzustand des Cursors
209–210	$D1–$D2	Zeiger auf den Anfang der Bildschirmzeile, auf welcher der Cursor gerade steht
211	$D3	Position des Cursors innerhalb einer logischen Zeile
214	$D6	Nummer der echten Zeile, in der sich der Cursor gerade befindet
647	$287	Zeichenfarbe unter dem Cursor
Bildschirm-Farbe		
243–244	$F3–$F4	Position des Cursors im Farbspeicher
646	$286	Aktuelle Farbe der Zeichen (Vordergrundfarbe)
647	$287	Zeichenfarbe unter dem Cursor
Bildschirm-Zeichen		
199	$C7	Flagge für reverse Darstellung der Zeichen
206	$CE	Bildschirmcode des Zeichens unter dem Cursor
212	$D4	Flagge für Gänsefuß-Modus
215	$D7	Zwischenspeicher für den ASCII-Codewert der zuletzt gedrückten Taste
216	$D8	Flagge für Insert-Modus
Bildschirm-Zeilen		
200	$C8	Zeiger auf das Ende der eingegebenen logischen Zeile
201-202	$C9–$CA	Zeiger auf Zeilen- und Spaltenposition des letzten Zeichens einer Zeile
209–210	$D1–$D2	Zeiger auf den Anfang der Bildschirmzeile, auf welcher der Cursor gerade steht

Adresse	Hex	Bedeutung
211	$D3	Position des Cursors innerhalb einer logischen Zeile
213	$D5	Länge einer Bildschirmzeile
214	$D6	Nummer der echten Zeile, auf der sich der Cursor gerade befindet
217–242	$D9–$F2	Link-Tabellen der Bildschirm-Zeilen
658	$292	Flagge für Scrollen
DATA		(siehe READ)
Datei		
152	$98	Anzahl der offenen Dateien
153	$99	Nummer des Eingabegerätes
154	$9A	Nummer des Ausgabegerätes
183	$B7	Länge des derzeitigen Datei-Namens
184	$B8	Nummer der derzeitigen Datei
185	$B9	Derzeitige Sekundär-Adresse
186	$BA	Derzeitige Geräte-Nummer
187–188	$BB–$BC	Zeiger auf Adresse des derzeitigen Datei-Namens
601–610	$259–$262	Tabelle der Datei-Nummern
611–620	$263–$26C	Tabelle der Geräte-Nummern
621–630	$26D–$276	Tabelle der Sekundär-Adressen
DIM		
11	$B	Anzahl der Dimensionen von Feldern (Arrays)
12	$C	Flagge für Basic-Routinen, die ein Feld suchen, beziehungsweise aufbauen
Eingabe-Puffer		
7	$7	Suchzeichen zur Prüfung von Basic-Texteingabe
8	$8	Suchzeichen speziell für Befehlsende und Gänsefüße
11	$B	Flagge für den Eingabe-Puffer
512–600	$200–$258	Eingabe-Puffer von Basic
Einschalten/Reset (beeinflußte Adressen)		
0–2	$0–$2	Sprungbefehl und wählbare Sprungadresse beim USR-Befehl (nur VC 20)
3–4	$3–$4	Vektor auf die Routine zur Umwandlung einer Gleitkommazahl in eine ganze Zahl mit Vorzeichen
5–6	$5–$6	Vektor auf die Routine zur Umwandlung einer ganzen Zahl in eine Gleitkommazahl
19	$13	Flagge zur Kennzeichnung des laufenden Ein-/Ausgabegerätes
22	$16	Zeiger auf freien Speicherplatz im String Descriptor Stack
43–44	$2B–$2C	Zeiger auf Anfang der Basic-Programme im Speicher
45–46	$2D–$2E	Zeiger auf Anfang der Variablen im Speicher (nur bei Reset)
51–52	$33–$34	Zeiger auf die untere Grenze des Speicherbereichs für den Text der Zeichenketten-Variablen
55–56	$37–$38	Zeiger auf das Ende des für Basic-Programme verfügbaren Speichers
122–123	$7A–$7B	Teil der CHRGET-Routine
139–143	$8B–$8F	Wert der RND-Funktion als Gleitkommazahl
153	$99	Nummer des Eingabe-Gerätes
154	$9A	Nummer des Ausgabe-Gerätes
160–162	$A0–$A2	Interne Uhr für TI und TI$ (nur beim Einschalten)
178–179	$B2–$B3	Zeiger auf den Kassetten-Puffer
195–196	$C3–$C4	Zeiger auf den Anfang des Programms hinter dem Tape Header
256–511	$100–$1FF	Stapelspeicher (Stack)
641–642	$281–$282	Zeiger auf den Anfang des Programmspeichers
643–644	$283–$284	Zeiger auf das Ende des Programmspeichers
646	$286	Aktuelle Farbe der Zeichen (Vordergrundfarbe)
648	$288	Beginn des Bildschirmspeichers
655–656	$28F–$290	Vektor auf die Routine der Tastencode-Tabellen
784–786	$310–$312	nur C64, identisch mit 0–3 beim VC 20
788–819	$314–$333	Indirekte Sprungvektoren auf Routinen des Betriebssystems
END		
57–58	$39–$3A	Nummer der laufenden Basic-Programmzeile
59–60	$3B–$3C	Zeilennummer der letzten Programmunterbrechung
61–62	$3D–$3E	Zeiger auf die Adresse, ab welcher der Text der laufenden Basic-Zeile abgespeichert ist
Felder (Arrays)		
11	$B	Anzahl der Dimensionen von Feldern (Arrays)
12	$C	Flagge für Basic-Routinen, die ein Feld suchen, beziehungsweise aufbauen
16	$10	Flagge zur Anzeige eines Variablenfeldes oder einer selbstdefinierten Funktion
47–48	$2F–$30	Zeiger auf die Anfangsadresse des Speicherbereiches für Felder (Arrays)
49–50	$31–$32	Zeiger auf die Endadresse des Speicherbereichs für Felder (Arrays)
FN		
16	$10	Flagge zur Anzeige eines Variablenfeldes oder einer selbstdefinierten Funktion
78–79	$4E–$4F	Zeiger auf Adresse, ab welcher der Wert der Variablen einer selbstdefinierten Funktion gespeichert ist
FOR-NEXT		
47–48	$2F–$30	Zeiger auf die Anfangsadresse des Speicherbereichs für Felder (Arrays)
57–58	$39–$3A	Nummer der laufenden Basic-Programmzeile
73–74	$49–$4A	Zwischenspeicher für Variable einer FOR-NEXT-Schleife
FRE		
49–50	$31–$32	Zeiger auf die Endadresse des Speicherbereichs für Felder (Arrays)
51–52	$33–$34	Zeiger auf die untere Grenze des Speicherbereichs für den Text der Zeichenketten-Variablen
Garbage Collection		
15	$F	Flagge bei LIST, Garbage Collection und Textumwandlung
49–50	$31–$32	Zeiger auf die Endadresse des Speicherbereichs für Felder (Arrays)
51–52	$33–$34	Zeiger auf die untere Grenze des Speicherbereichs für den Text der Zeichenketten-Variablen
83	$53	Flagge für Garbage Collection
GET		
17	$11	Flagge für INPUT, GET oder READ
18	$13	Flagge zur Kennzeichnung des laufenden Ein-/Ausgabegerätes
67–68	$43–$44	Zeiger auf die Adresse, aus welcher die Befehle INPUT, GET und READ die Zeichen/Zahlen holen
GET #		
19	$13	Flagge zur Kennzeichnung des laufenden Ein-/Ausgabegerätes
153	$99	Nummer des Eingabegerätes
Gleitkomma		
3–4	$3–$4	Vektor auf die Routine zur Umwandlung einer Gleitkommazahl in eine ganze Zahl mit Vorzeichen
5–6	$5–$6	Vektor auf die Routine zur Umwandlung einer ganzen Zahl in eine Gleitkommazahl
97–102	$61–$66	Gleitkomma-Akkumulator Nr. 1
104	$68	Überlauf-Speicher des Gleitkomma-Akkumulators Nr.1
105–110	$69–$6E	Gleitkomma-Akkumulator Nr. 2
111	$6F	Flagge für Vorzeichenvergleich der Gleitkomma-Akkumulatoren 1 und 2
112	$70	Rundungsspeicher des Gleitkomma-Akkumulators Nr.1
255	$FF	Zwischenspeicher von Gleitkommazahlen in ASCII-Werte
256–266	$100–$10A	Arbeitsspeicher für Umwandlung von Gleitkommazahlen in ASCII-Werte
778–779	$30A–$30B	Indirekter Sprungvektor auf die Basic-Routine, die einen numerischen Ausdruck in eine Gleitkommazahl umwandelt
GOTO		
20–21	$14–$15	Zeilennummer für LIST, GOTO, GOSUB und ON
57–58	$39–$3A	Nummer der laufenden Basic-Programmzeile
INPUT		
17	$11	Flagge für INPUT, GET oder READ
67–68	$43–$44	Zeiger auf die Adresse, aus welcher die Befehle INPUT, GET und READ die Zeichen/Zahlen holen
INPUT #		
19	$13	Flagge zur Kennzeichnung des laufenden Ein-/Ausgabegerätes
153	$99	Nummer des Eingabegerätes
INST		
212	$D4	Flagge für Gänsefuß-Modus
216	$D8	Flagge für INSERT-Modus
Interrupt mit BREAK		
57–58	$39–$3A	Nummer der laufenden Basic-Programmzeile
170	$AA	Zwischenspeicher für Kassettenroutinen
663	$297	RS232-Status-Register
790–791	$316–$317	Vektor auf die BREAK-Interrupt-Routine
Interrupt mit IRQ		
671–672	$29F–$2A0	Zwischenspeicher für den IRQ-Vektor während Kassetten-Ein-/Ausgabe
788–789	$314–$315	Vektor auf die IRQ-Interrupt-Routine
Interrupt mit NMI		
792–793	$318–$319	Vektor auf die NMI-Interrupt-Routine
Kassettenpuffer		
166	$A6	Zähler der bearbeiteten Bytes im Kassettenpuffer
178–179	$B2–$B3	Zeiger auf den Kassettenpuffer
828–1019	$33C–$3FB	Kassettenpuffer
LIST		
15	$F	Flagge bei LIST, Garbage Collection und Textumwandlung
20–21	$14–$15	Zeilennummer für LIST, GOTO, GOSUB und ON
LOAD/VERIFY		
10	$A	Flagge für LOAD oder VERIFY
147	$93	Flagge für LOAD oder VERIFY
172–173	$AC–$AD	Zeiger auf die Anfangsadresse für Ein-/Ausgabe
174–175	$AE–$AF	Zeiger auf die Endadresse für Ein-/Ausgabe
183	$B7	Länge des File-Namens
185	$B9	Sekundär-Adresse
187–188	$BB–$BC	Zeiger auf Adresse des derzeitigen File-Namens

195–196	$C3–$C4	Zeiger auf den Anfang des Programms hinter dem Tape Header
816–817	$330–$331	Indirekter Sprungvektor auf die LOAD-Routine des Betriebssystems
NEXT (siehe FOR)		
READ DATA		
17	$11	Flagge für INPUT, GET oder READ
63–64	$3F–$40	Zeilennummer des gerade laufenden DATA-Befehls
65–66	$41–$42	Zeiger auf die Adresse, ab der die laufenden DATA-Angaben gespeichert sind
67–68	$43–$44	Zeiger auf die Adresse, aus welcher die Befehle INPUT, GET und READ die Zeichen/Zahlen holen
75–76	$4B–$4C	Zwischenspeicher für Zeiger bei READ und bei mathematischen Operationen
RESET (siehe Einschalten)		
RND		
139–143	$8B–$8F	Wert der RND-Funktion als Gleitkommazahl
RS232-Schnittstelle		
167	$A7	Zwischenspeicher für Eingabe über die RS232-Schnittstelle
168	$A8	Bitzähler für RS232-Eingabe
169	$A9	RS232-Flagge für Startbit-Prüfung
170	$AA	RS232-Eingabespeicher
171	$AB	Paritypрüfung
181	$B5	RS232-Anzeige für nächstes Bit
182	$B6	Ausgabe-Zwischenspeicher für RS232
189	$BD	Zwischenspeicher für RS232-Parity-Prüfung
247–248	$F7–$F8	Zeiger auf den Anfang des RS232-Eingabepuffers
249–250	$F9–$FA	Zeiger auf den Anfang des RS232-Ausgabepuffers
659	$293	RS232-Steuerregister
660	$294	RS232-Befehlsregister
661–662	$295–$296	RS232 frei wählbare Baudrate
663	$297	RS232-Statusregister
664	$298	RS232-Anzahl der zu übertragenden Bits
665–666	$299–$29A	Zeit, die bei RS232 zum Übertragen eines Bits gebraucht wird
667	$29B	Index auf das Ende des RS232-Eingabepuffers
668	$29C	Index auf den Anfang des RS232-Eingabepuffers
669	$29D	Index auf den Anfang des RS232-Ausgabepuffers
670	$29E	Index auf das Ende des RS232-Ausgabepuffers
SAVE		
172–173	$AC–$AD	Zeiger auf die Anfangsadresse für Ein-/Ausgabe
174–175	$AE–$AF	Zeiger auf die Endadresse für Ein-/Ausgabe
818–819	$332–$333	Indirekter Sprungvektor auf die SAVE-Routine
Serielle Schnittstelle		
148	$94	Flagge für Floppy/Drucker-Ausgabe
149	$94	Zeichen im Ausgabepuffer
163–164	$A3–$A4	Zwischenspeicher
172–173	$AC–$AD	Zeiger auf die Anfangsadresse für Ein-/Ausgabe
174–175	$AE–$AF	Zeiger auf die Endadresse für Ein-/Ausgabe
193–194	$C1–$C2	Anfangsadresse für Ein-/Ausgabe-Operationen
SIN		
18	$12	Flagge für Vorzeichen des Ergebnisses bei SIN und TAN
Speicherbelegung		
43–44	$2B–$2C	Zeiger auf den Anfang der Basic-Programme im Speicher
45–46	$2D–$2E	Zeiger auf die Anfangsadresse des Speicherbereichs für Variable
47–48	$2F–$30	Zeiger auf die Anfangsadresse des Speicherbereichs für Felder (Arrays)
49–50	$31–$32	Zeiger auf die Endadresse +1 des Speicherbereichs für Felder (Arrays)
51–52	$33–$34	Zeiger auf die untere Grenze des Speicherbereichs für den Text der String-Variablen
53–54	$35–$36	Zeiger auf die Adresse des zuletzt eingegebenen Strings
55–56	$37–$38	Zeiger auf das Ende des für Basic-Programme verfügbaren Speichers
641–642	$281–$282	Zeiger auf den Anfang des Programmspeichers
643–644	$283–$284	Zeiger auf das Ende des Programmspeichers
648	$288	Beginn des Bildschirmspeichers
Speicher zur freien Verfügung		
146–150	$92–$96	nur wenn Datasette nicht benutzt wird
163–177	$A3–$B1	nur wenn Datasette oder RS232-Schnittstelle nicht benutzt wird
247–250	$F7–$FA	nur wenn RS232-Schnittstelle nicht benutzt wird
251–254	$FB–$FE	
659–670	$293–$29E	nur wenn RS232-Schnittstelle nicht benutzt wird
671–672	$29F–$2A0	nur wenn Datasette nicht benutzt wird
673–678	$2A1–$2A6	nur beim VC 20
679–767	$2A7–$2FF	
784–787	$310–$313	nur beim VC 20
820–827	$334–$33B	
828–1019	$33C–$3FB	nur wenn Datasette nicht benutzt wird
1020–1023	$3FC–$3FF	
ST(atus)		
144	$90	Status-Variable ST
663	$297	RS232-Statusregister
Stapelspeicher (Stack)		
25–33	$19–$21	Stack für vorläufige Zeichenketten
319–511	$13F–$1FF	Speicherbereich des Mikroprozessor-Stapels
STOP		
57–58	$39–$3A	Nummer der laufenden Basic-Programmzeile
59–60	$3B–$3C	Zeilennummer der letzten Programmunterbrechung
145	$91	Zwischenspeicher für Abfrage der STOP-Taste
808–809	$328–$329	Indirekter Sprungvektor auf die STOP-Routine des Betriebssystems
STRINGS		
22	$16	Zeiger auf freien Speicherplatz im String Descriptor Stack
23–24	$17–$18	Zeiger auf die Adresse der letzten Zeichenkette im Temporary String Stack
25–33	$19–$21	Descriptor Stack für vorläufige Zeichenketten
51–52	$33–$34	Zeiger auf die untere Grenze des Speicherbereichs für den Text der String-Variablen
53–54	$35–$36	Zeiger auf die Adresse des zuletzt eingegebenen Strings
80–82	$50–$52	Zeiger auf einen vorläufigen Speicherplatz einer Zeichenkette, die gerade bearbeitet wird
SYS		
780	$30C	Speicher für den Akkumulator
781	$30D	Speicher für das X-Register
782	$30E	Speicher für das Y-Register
783	$30F	Speicher für das Status-(P)-Register
TAN		
18	$12	Flagge für Vorzeichen des Ergebnisses bei SIN und TAN










Comic: Ingo Stein



Checksummer V3 und MSE

Diese beiden Programme sind unentbehrlich beim Abtippen unserer Listings. Sie helfen, Tippfehler vor allem bei Maschinenprogrammen zu vermeiden und sparen eine Menge Zeit.

Nobody is perfect. Jeder Computer-Fan, egal ob blutiger Anfänger oder ausgefuchster Profi, macht beim Abtippen von Programmen Tippfehler. Diese Fehler später zu finden, kann ein langwieriges Unterfangen sein.

Deshalb haben wir für Sie die Programme »Checksummer V3« und »MSE« (MaschinenSpracheEditor) entwickelt. Der Checksummer ist für Basic-Programme und der MSE für Maschinensprache-Listings zuständig.

Der Checksummer

Zuerst einmal müssen Sie das Checksummer-Programm (siehe Listing 1) abtippen. Dabei sollten Sie äußerst sorgfältig vorgehen, vor allem bei den Zahlen in den DATA-Zeilen 20 bis 30. Wenn Sie trotzdem noch einen Tippfehler gemacht haben, meldet sich das Programm später mit einem entsprechenden Hinweis. Wenn Sie fertig sind, speichern Sie das Programm auf Diskette oder Kassette.

Jetzt geht es los:

1. Starten Sie den Checksummer durch die Eingabe von »RUN« und das Drücken der RETURN-Taste.
2. Wenn die Meldung »Checksummer aktiviert...« auf dem Bildschirm erscheint, haben Sie keinen Tippfehler gemacht und der Checksummer ist nun eingeschaltet.
3. Zum Löschen des Basic-Programms geben Sie bitte »NEW« ein. Keine Angst, der Checksummer selbst wird dadurch nicht gelöscht.
4. Nun können wir den Checksummer testen. Geben Sie bitte folgende Zeile ein und drücken Sie die RETURN-Taste:

1 REM

In der linken oberen Bildschirmecke sehen Sie nun die Prüfsumme über die eben eingegebene Basic-Zeile. Sie muß <63> lauten. Dem Checksummer ist es übrigens egal, ob Sie »1 REM« oder »1REM« eintippen. Nur innerhalb von Anführungszeichen ist die richtige Anzahl an Leerzeichen wichtig. Diese Prüfsummen erscheinen (sofern Sie den Checksummer eingeschaltet haben) immer dann, wenn Sie eine Basic-Zeile eintippen und dann die RETURN-Taste drücken. In der 64'er finden Sie die Prüfsumme immer am Ende jeder Programmzeile.

```
10 PRINT"CHECKSUMMER FUER C 64"
11 PRINT:PRINT"EINEN MOMENT, BITTE ..."
12 FOR I=828 TO 864:READ A:POKE I,A:PS=PS+
   A:NEXT I
13 IF PS<>5765 THEN PRINT"TIPPFEHLER IN DE
   N ZEILEN 20 BIS 22":END
14 SYS 828:PS=0:FOR I=58464 TO 58583:READ
   A:POKE I,A:PS=PS+A:NEXT I
15 IF PS<>16147 THEN PRINT"TIPPFEHLER IN D
   EN ZEILEN 22 BIS 30":END
16 POKE 1,53:POKE 42289,96:POKE 42290,228
17 PRINT"CHECKSUMMER AKTIVIERT."
18 PRINT:PRINT" AUSSCHALTEN : POKE1,55 ODE
   R"SPC(27)"<RUN/STOP+RESTORE>"
19 PRINT:PRINT" ANSCHALTEN : POKE1,53"
20 DATA 169,0,133,254,162,1,189,93,3,133,2
   55,160,0,177,254
21 DATA 145,254,136,208,249,230,255,165,25
   5,221,95,3,208,238,202
22 DATA 16,230,96,160,224,192,0,160,2,169,
   0,170,133,254,177
23 DATA 95,240,40,201,32,208,3,200,208,245
   ,133,255,138,41,7
24 DATA 170,240,14,72,165,255,24,42,105,0,
   202,208,249,133,255
25 DATA 104,170,232,165,255,24,101,254,133
   ,254,76,111,228,192,4
26 DATA 48,219,198,214,165,214,72,162,3,16
   9,32,157,1,4,189
27 DATA 212,228,32,210,255,208,12,0,92,72,
   32,201,255,170,104
28 DATA 144,1,138,96,202,16,228,166,254,16
   9,0,32,205,189,169
29 DATA 62,32,210,255,104,133,214,32,108,2
   29,169,141,32,210,255
30 DATA 76,128,164,9,60,18,19
© 64'er
```

Listing 1. Der »Checksummer 64 V3« für Basic-Listings

```
5 PRINT CHR$(14)                              <242>
10 PRINT"{CLR}"                               <254>
20 PRINT"******************************"      <130>
30 PRINT"{4DOWN,2SPACE}TEST{SPACE,BLUE,6SP
   ACE}"                                      <022>
40 PRINT"BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"      <108>
© 64'er
```

Bild 1. Die Bedeutung der Steuerzeichen wird im nachfolgenden Text erklärt

In Zeile 10 müssen Sie nach den Anführungszeichen die Tasten <SHIFT CLR/HOME> drücken und nicht die Klammern mit dem Wort CLR eingeben. In Zeile 20 drücken Sie nach den Anführungszeichen die CBM-Taste und den Buchstaben <Q>, gefolgt von mehreren SHIFT- und Stern-Tasten und zum Schluß die CBM-Taste und den Buchstaben <W>. In Zeile 30 ist es viermal die CURSOR-abwärts-Taste, gefolgt von zweimaliger Leertaste, dann <SHIFT T> und normal EST, zum Schluß noch einmal die Leertaste, die Farbtaste Blau <CTRL 7> und sechsmal die Leertaste. Zeile 40 besteht lediglich aus mehreren Grafikzeichen, die mit der CBM-Taste und <B> erzeugt werden.

CTRL steht für Control-Taste, so bedeutet {CTRL+A}, daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

{DOWN}	Taste neben rechtem Shift, Cursor unten	{SPACE}	Leertaste	{RVSON}	Control-Taste & 9
{UP}	Shift-Taste & Taste neben rechtem Shift; Cursor hoch	{SHIFT-Space}	Shift-Taste & Leertaste	{RVOFF}	Control-Taste & 0
{CLR}	Shift-Taste & 2. Taste ganz rechts oben	{F1} bis {F8}	Funktionstasten	{ORANGE}	Commodore-Taste & 1
{INST}	Shift-Taste & Taste ganz rechts oben	{RETURN}	Return-Taste	{BROWN}	Commodore-Taste & 2
{HOME}	2. Taste von ganz rechts oben	{BLACK}	Control-Taste & 1	{LIG.RED}	Commodore-Taste & 3
{DEL}	Taste ganz rechts oben	{WHITE}	Control-Taste & 2	{GREY 1}	Commodore-Taste & 4
{RIGHT}	Taste ganz rechts unten	{RED}	Control-Taste & 3	{GREY 2}	Commodore-Taste & 5
{LEFT}	Shift-Taste & Taste unten rechts	{CYAN}	Control-Taste & 4	{LIG.GREEN}	Commodore-Taste & 6
		{PURPLE}	Control-Taste & 5	{LIG.BLUE}	Commodore-Taste & 7
		{GREEN}	Control-Taste & 6	{GREY 3}	Commodore-Taste & 8
		{BLUE}	Control-Taste & 7		
		{YELLOW}	Control-Taste & 8		

Tabelle 1. Die Steuerbefehle in den Listings

Diese Zahlen dürfen Sie NICHT mit abtippen.
Als Beispiel sehen Sie Bild 1. Am rechten Rand jeder Spalte sehen Sie die Prüfsummen in eckigen Klammern.

Damit sind wir beim zweiten wichtigen Punkt: Sehen Sie sich die Zeile 240 von Listing 2 genauer an. Nach dem ersten Anführungszeichen nach dem PRINT-Befehl sehen Sie eine geschweifte Klammer { }. Immer, wenn Sie in einem unserer Listings diese Klammern sehen, dürfen Sie das, was innerhalb der Klammern steht, nicht eintippen. Sie müssen die entsprechende Taste drücken. Beispiel:
10 PRINT "{CLR}"
bedeutet: Nach dem Anführungszeichen die »Bildschirmlöschen«-Taste drücken (<SHIFT CLR/HOME>). In Tabelle 1 sehen Sie eine Zusammenfassung aller möglichen Steuertasten mit dem entsprechenden Klartext.

Weiterhin sehen Sie in Bild 1 (Bedeutung der Steuerzeichen) in Zeile 30 ein unterstrichenes »T« nach der Klammer. Das bedeutet, daß Sie ein »T« zusammen mit der SHIFT-Taste drücken müssen, also <SHIFT T>. Wenn ein Zeichen »überstrichen« ist, müssen Sie dieses zusammen mit der CBM-Taste eingeben. Die CBM-Taste befindet sich ganz links unten auf der Tastatur und hat die Aufschrift »C=«.

```
100 REM DIESES PROGRAMM ERZEUGT DEN               <210>
110 REM MSE V1.1 AUF DISKETTE.                     <039>
120 REM BESITZER EINER DATASETTE                   <178>
130 REM MUESSEN DIE '8' AM ENDE VON                <145>
140 REM ZEILE 343 IN EINE '1' AENDERN!             <176>
150 REM                                            <212>
230 IF PEEK(44)<>32 THEN PRINT"{CLR}SIE HA
    BEN VERGESSEN, DIE POKES EINZUGE- BEN!
    ":END                                          <050>
240 PRINT"{CLR}";:DIM H(75):FOR I=0 TO 9           <042>
250 H(48+I)=I:H(65+I)=I+10:NEXT:Z=1000             <136>
260 FOR I=2048 TO 3755 STEP 20:PRINT"{HOME
    }ICH LESE ZEILE:"Z                             <253>
261 FOR N=0 TO 19:READ A$:IF LEN(A$)<>2 TH
    EN 900                                         <062>
262 IF PEEK(63)+PEEK(64)*256<>Z THEN 800           <011>
270 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))         <199>
280 D=H(H)*16+H(L):S=S+D:POKE I+N,D                <165>
290 NEXT:READ V:IF S<>V THEN 900                   <139>
300 S=0:Z=Z+1:NEXT:R=PEEK(2111):H=PEEK(210
    6)                                             <126>
301 POKE 53280,R:POKE 53281,H:POKE 646,R:P
    RINT"{CLR}DIE DATA-ZEILEN SIND FEHLERF
    REI!"                                          <080>
302 PRINT"SIE KOENNEN NUN DIE FARBEN DES M
    SE"                                            <209>
303 PRINT"EINSTELLEN.":PRINT"{2DOWN,SPACE,
    RVSON}DRUECKEN SIE <1>, <2> ODER <9>           <205>
304 PRINT"{DOWN,2SPACE}<1> - RAHMEN-/SCHRI
    FTFARBE                                        <013>
305 PRINT"{2SPACE}<2> - HINTERGRUNDFARBE           <233>
306 PRINT"{DOWN,2SPACE}<9> - FARBEN UEBERN
    EHMEN                                          <158>
307 PRINT"{2DOWN}FARBE <1>:"R:PRINT"FARBE
    <2>:"H                                         <066>
308 GET A:IF A=0 THEN 308                          <210>
309 IF A=1 THEN R=(R+1)AND 15                      <098>
310 IF A=2 THEN H=(H+1)AND 15                      <086>
311 IF A=9 THEN 340                                <217>
312 GOTO 301                                       <034>
340 POKE 2106,H:POKE 2111,R                        <153>
342 POKE 631,19:POKE 632,13:POKE 198,2             <135>
343 PRINT"{CLR}SAVE"CHR$(34)"MSE V1.1"CHR$
    (34)",8                                        <091>
344 POKE 43,1:POKE 44,8:POKE 45,172:POKE 4
    6,14:END                                       <140>
800 PRINT"{CLR,RVSON}SIE HABEN ZEILE"Z"{LE
    FT,SPACE}VERGESSEN:":A=PEEK(646)AND 15         <124>
810 POKE 646,PEEK(53281)AND 15:PRINT"LIST"
    Z-2"-"Z+2:POKE 646,A                           <224>
820 GOTO 920                                       <082>
900 PRINT"{CLR,RVSON}SIE HABEN EINEN TIPPF
    EHLER GEMACHT:":A=PEEK(646)AND 15              <154>
910 POKE 646,PEEK(53281)AND 15:PRINT"LIST"
    Z:POKE 646,A                                   <173>
920 POKE 631,19:POKE 632,17:POKE 633,13:PO
    KE 198,3:END                                   <126>
1000 DATA 00,0B,08,0A,00,9E,32,30,36,31,00
     ,00,00,A2,08,A9,36,85,A4,A9, 1247             <119>
1001 DATA 08,85,A5,A9,00,85,A6,A9,B0,85,A7
     ,A0,00,B1,A4,91,A6,C8,D0,F9, 2888             <054>
1002 DATA E6,A5,E6,A7,CA,D0,F2,A9,36,85,01
     ,4C,00,B0,20,D1,B1,A9,00,8D, 2781             <096>
1003 DATA 21,D0,A9,0F,8D,20,D0,8D,86,02,A0
     ,B3,A9,74,20,FF,B1,A0,B3,A9, 2679             <089>
1004 DATA B9,20,FF,B1,A0,00,20,CF,FF,99,01
     ,02,C8,C9,0D,D0,F5,88,F0,D2, 2912             <217>
1005 DATA C0,11,90,02,A0,10,8C,00,02,20,EA
     ,B1,A0,B3,A9,CF,20,FF,B1,20, 2327             <045>
1006 DATA 8E,B4,85,FC,85,62,20,8E,B4,85,FB
     ,85,61,20,A7,B4,D0,20,A0,B3, 2864             <199>
1007 DATA A9,E5,20,FF,B1,20,8E,B4,85,60,20
     ,8E,B4,85,5F,20,A7,B4,D0,0A, 2624             <091>
1008 DATA A5,61,C5,5F,A5,62,E5,60,90,06,20
     ,43,B3,4C,3A,B0,A9,AA,A0,00, 2379             <167>
1009 DATA EA,EA,E6,FB,D0,02,E6,FC,20,3F,B2
     ,90,EF,4C,FB,B4,A2,02,86,58, 3190             <041>
1010 DATA A9,A6,A0,9D,20,F2,B1,20,E4,FF,F0
     ,FB,C9,30,90,0C,C9,47,B0,08, 2970             <231>
1011 DATA C9,3A,90,0B,C9,41,B0,07,C9,14,D0
     ,0F,4C,0B,B1,20,D2,FF,A6,58, 2322             <121>
1012 DATA 95,F7,C6,58,D0,D2,60,AE,8D,02,F0
     ,26,C9,0C,D0,03,4C,0B,B6,C9, 2685             <057>
1013 DATA 13,D0,03,4C,8B,B5,C9,0D,D0,03,4C
     ,BA,B4,C9,10,D0,03,4C,68,B5, 2282             <225>
1014 DATA C9,0E,D0,06,20,5F,B4,4C,64,B1,4C
     ,92,B0,A5,F9,20,02,B1,0A,0A, 2132             <208>
1015 DATA 0A,0A,85,F9,A5,F8,20,02,B1,05,F9
     ,60,C9,3A,90,02,69,08,29,0F, 1950             <092>
1016 DATA 60,A6,59,E0,08,90,1F,A6,58,E0,02
     ,B0,06,20,D2,FF,4C,8E,B0,C6, 2509             <188>
1017 DATA 59,A0,14,A9,92,20,F2,B1,CA,D0,FA
     ,84,57,68,68,4C,8B,B1,A6,D3, 2891             <197>
1018 DATA E0,08,B0,03,4C,92,B0,20,D2,FF,A6
     ,58,E0,02,90,09,C6,59,20,D2, 2468             <049>
1019 DATA FF,C6,58,D0,F9,4C,8E,B0,48,4A,4A
     ,4A,4A,20,59,B1,68,29,0F,C9, 2419             <035>
1020 DATA 0A,90,02,69,06,69,30,4C,D2,FF,A2
     ,FC,9A,20,D1,B1,20,48,B2,20, 2261             <073>
1021 DATA EA,B1,20,9F,B2,A5,FC,20,4E,B1,A5
     ,FB,20,4E,B1,20,ED,B1,A9,3A, 2860             <148>
1022 DATA A0,20,20,F2,B1,A9,00,85,59,20,8E
     ,B0,20,ED,B1,A4,59,20,EF,B0, 2530             <233>
1023 DATA 91,FB,C8,84,59,C0,08,90,EC,20,10
     ,B2,A9,12,20,D2,FF,20,8E,B0, 2657             <105>
1024 DATA 20,EF,B0,C5,FF,F0,0D,20,43,B3,A9
     ,14,A0,14,20,F2,B1,4C,A2,B1, 2665             <034>
1025 DATA A9,92,20,D2,FF,20,33,B2,20,E0,B2
     ,20,3F,B2,90,9F,4C,8B,B5,A9, 2648             <123>
1026 DATA 93,20,D2,FF,A2,00,A9,03,9D,00,D8
     ,9D,00,D9,9D,00,DA,9D,00,DB, 2476             <237>
1027 DATA E8,D0,EF,60,A9,0D,2C,A9,20,4C,D2
     ,FF,20,D2,FF,98,4C,D2,FF,20, 2965             <160>
1028 DATA E4,FF,F0,FB,60,84,5D,85,5C,A0,00
     ,B1,5C,F0,06,20,D2,FF,C8,D0, 3100             <077>
1029 DATA F6,60,A5,FB,85,5A,A0,00,84,5B,B1
     ,FB,18,65,5A,85,5A,90,02,E6, 2606             <156>
1030 DATA 5B,06,5A,26,5B,C8,C0,08,90,EC,A5
     ,5A,65,5B,85,FF,60,18,A5,FB, 2467             <219>
1031 DATA 69,08,85,FB,90,02,E6,FC,60,A5,FB
     ,C5,5F,A5,FC,E5,60,60,A0,B3, 3106             <183>
1032 DATA A9,FB,20,FF,B1,A0,01,B9,00,02,20
     ,D2,FF,CC,00,02,C8,90,F4,A9, 2692             <098>
1033 DATA 14,ED,00,02,AA,20,ED,B1,CA,D0,FA
     ,A5,62,20,4E,B1,A5,61,20,4E, 2457             <060>
1034 DATA B1,20,ED,B1,A5,60,20,4E,B1,A5,5F
     ,20,4E,B1,EA,EA,EA,EA,EA,EA, 3122             <190>
1035 DATA EA,EA,24,5E,10,01,60,A9,12,20,D2
     ,FF,A2,28,20,ED,B1,CA,D0,FA, 2703             <087>
1036 DATA A9,92,4C,D2,FF,A5,D6,C9,16,B0,01
     ,60,A9,A0,85,A4,A9,78,85,A6, 2945             <204>
1037 DATA A9,04,85,A5,85,A7,A2,13,A0,27,B1
     ,A4,91,A6,88,10,F9,CA,F0,19, 2671             <208>
1038 DATA 18,A5,A4,69,28,85,A4,90,02,E6,A5
     ,18,A5,A6,69,28,85,A6,90,E0, 2503             <251>
1039 DATA E6,A7,4C,B6,B2,A9,91,4C,D2,FF,A9
     ,0F,8D,18,D4,A9,00,8D,05,D4, 2776             <000>
1040 DATA A9,F7,8D,06,D4,A9,11,8D,04,D4,A9
     ,32,8D,01,D4,A9,00,8D,00,D4, 2413             <126>
1041 DATA A0,80,20,09,B3,A9,10,8D,04,D4,60
     ,A2,FF,CA,D0,FD,88,D0,F8,60, 2914             <240>
1042 DATA A9,0F,8D,18,D4,A9,2D,8D,05,D4,A9
     ,A5,8D,06,D4,A9,21,8D,04,D4, 2385             <119>
1043 DATA A9,07,8D,01,D4,A9,05,8D,00,D4,A0
```

```
       ,FF,20,09,B3,A9,20,8D,04,D4, 2250     <078>
1044 DATA A9,00,8D,01,D4,8D,00,D4,60,38,20
       ,F0,FF,8A,48,98,48,18,A0,06, 2179     <175>
1045 DATA A2,18,20,F0,FF,A0,B4,A9,0A,20,FF
       ,B1,20,12,B3,20,E4,FF,F0,FB, 2931     <093>
1046 DATA A2,1D,A9,14,20,D2,FF,CA,D0,FA,68
       ,A8,68,AA,18,4C,F0,FF,0D,0D, 2704     <088>
1047 DATA 0D,20,20,20,20,20,20,20,4D,41,53
       ,43,48,49,4E,45,4E,53,50,52, 1144     <216>
1048 DATA 41,43,48,45,20,2D,20,45,44,49,54
       ,4F,52,20,0D,0D,20,20,20,20, 1023     <038>
1049 DATA 20,20,20,20,56,4F,4E,20,4E,2E,4D
       ,41,4E,4E,20,26,20,44,2E,57, 1128     <206>
1050 DATA 45,49,4E,45,43,4B,00,0D,0D,0D,20
       ,20,20,50,52,4F,47,52,41,4D, 1102     <117>
1051 DATA 4D,4E,41,4D,45,20,3A,20,00,0D,0D
       ,20,20,20,53,54,41,52,54,41, 1073     <095>
1052 DATA 44,52,45,53,53,45,20,3A,20,24,00
       ,0D,0D,20,20,20,45,4E,44,41, 1014     <129>
1053 DATA 44,52,45,53,53,45,20,20,20,3A,20
       ,24,00,92,01,01,50,52,4F,47, 1136     <228>
1054 DATA 52,41,4D,4D,20,3A,20,00,12,20,20
       ,2A,2A,2A,20,46,41,4C,53,43, 1024     <027>
1055 DATA 48,45,20,45,49,4E,47,41,42,45,20
       ,2A,2A,2A,20,20,92,00,0D,0D, 1058     <098>
1056 DATA 2A,2A,2A,20,45,4E,44,45,20,2A,2A
       ,2A,00,13,01,20,20,12,44,92, 916      <153>
1057 DATA 49,53,4B,20,4F,44,45,52,20,12,54
       ,92,41,50,45,0D,00,13,20,20, 1151     <035>
1058 DATA 49,2F,4F,20,2D,20,46,45,48,4C,45
       ,52,00,20,D1,B1,20,48,B2,A0, 1606     <012>
1059 DATA B3,A9,CF,20,FF,B1,20,8E,B4,85,FC
       ,20,8E,B4,85,FB,C5,61,A5,FC, 3207     <251>
1060 DATA E5,62,90,23,A5,FB,C5,5F,A5,FC,E5
       ,60,B0,19,20,A7,B4,D0,14,60, 2860     <112>
1061 DATA 20,A7,B4,F0,0C,85,F9,20,A7,B4,F0
       ,05,85,F8,4C,EF,B0,68,68,20, 2749     <088>
1062 DATA 43,B3,4C,5F,B4,20,CF,FF,C9,4C,D0
       ,09,20,D1,B1,20,48,B2,4C,0B, 2372     <046>
1063 DATA B6,C9,0D,60,A9,00,85,5E,20,5F,B4
       ,20,EA,B1,20,0D,B5,24,5E,30, 2042     <120>
1064 DATA 05,20,E4,FF,F0,FB,20,E1,FF,F0,26
       ,20,9F,B2,24,5E,10,09,20,4E, 2435     <198>
1065 DATA B5,20,0D,B5,20,60,B5,20,33,B2,20
       ,3F,B2,90,D7,A0,B4,A9,28,20, 2190     <207>
1066 DATA FF,B1,20,E4,FF,C9,0D,D0,F9,A9,00
       ,85,5E,A5,61,85,FB,A5,62,85, 3056     <240>
1067 DATA FC,20,E0,B2,4C,64,B1,A5,FC,20,4E
       ,B1,A5,FB,85,FF,20,4E,B1,A9, 3003     <221>
1068 DATA 20,A0,3A,20,F2,B1,A0,00,20,ED,B1
       ,B1,FB,20,4E,B1,C8,C0,08,90, 2566     <070>
1069 DATA F3,20,ED,B1,24,5E,30,03,A9,12,2C
       ,A9,20,20,D2,FF,20,10,B2,A5, 2190     <059>
1070 DATA FF,20,4E,B1,A9,92,20,D2,FF,4C,EA
       ,B1,A9,FF,85,B8,85,B9,A9,04, 3073     <029>
1071 DATA 85,BA,20,C0,FF,A2,FF,4C,C9,FF,20
       ,CC,FF,A9,FF,4C,C3,FF,20,5F, 3315     <189>
1072 DATA B4,A9,80,85,5E,20,4E,B5,20,48,B2
       ,A2,24,A9,2D,20,D2,FF,CA,D0, 2596     <111>
1073 DATA FA,20,EA,B1,20,EA,B1,20,60,B5,4C
       ,C1,B4,20,B8,B5,A6,5F,A4,60, 2812     <015>
1074 DATA A9,61,20,D8,FF,B0,0A,20,B7,FF,29
       ,BF,D0,03,4C,FB,B4,A9,01,20, 2577     <201>
1075 DATA C3,FF,20,68,B6,A0,B4,A9,4F,20,FF
       ,B1,20,F9,B1,4C,FB,B4,20,68, 2921     <237>
1076 DATA B6,A9,37,A0,B4,20,FF,B1,20,F9,B1
       ,A2,08,C9,44,F0,06,A2,01,C9, 2717     <213>
1077 DATA 54,D0,F1,A9,01,A8,20,BA,FF,A0,00
       ,E0,01,F0,1A,A9,40,8D,20,02, 2403     <101>
1078 DATA A9,3A,8D,21,02,B9,01,02,99,22,02
       ,C8,CC,00,02,90,F4,C8,C8,D0, 2182     <127>
1079 DATA 0C,B9,01,02,99,20,02,C8,CC,00,02
       ,D0,F4,98,A2,20,A0,02,4C,BD, 2018     <025>
1080 DATA FF,20,B8,B5,A5,BA,C9,08,90,33,A6
       ,B9,86,57,A9,01,20,C3,FF,A9, 2800     <022>
1081 DATA 60,85,B9,20,C0,FF,B0,28,A5,BA,20
       ,B4,FF,A5,B9,20,96,FF,20,A5, 2911     <053>
1082 DATA FF,85,61,A5,90,4A,4A,B0,13,20,A5
       ,FF,85,62,20,AB,FF,A5,57,85, 2663     <214>
1083 DATA B9,A9,00,20,D5,FF,90,03,4C,A3,B5
       ,86,5F,84,60,A5,BA,C9,01,D0, 2639     <131>
1084 DATA 0A,AD,3D,03,85,61,AD,3E,03,85,62
       ,4C,FB,B4,A9,13,20,D2,FF,A2, 2300     <120>
1085 DATA 1C,20,ED,B1,CA,D0,FA,60,00,00,00
       ,00,00,00,00,00,00,00,00,00, 1230     <143>
```


Listing 2. Der MSE-Lader

Der MSE

Der MSE dient zur Eingabe von Maschinensprache-Programmen. Als erstes müssen Sie den sogenannten »MSE-Lader« (Listing 2) abtippen. Dieser erzeugt erst das eigentliche MSE-Programm auf Diskette oder Kassette.

Wichtig: Vor dem Eintippen des MSE-Laders müssen Sie unbedingt ein paar Befehle eingeben (ohne Basic-Zeilennummer): POKE 44,32 : POKE 8192,0 : NEW

Jetzt können Sie beginnen, das Listing 2 abzutippen. Der MSE-Lader erkennt zwar, wenn Sie beim Eintippen der DATA-Zeilen einen Fehler gemacht haben, aber wenn Sie ganz sicher gehen möchten, sollten Sie den Checksummer vor dem Eintippen aktivieren. Die Prüfsummen für den MSE-Lader finden Sie am Ende der jeweiligen Programmzeilen.

Wenn Sie das Listing 2 nicht auf einmal abtippen möchten, müssen Sie vor jedem neuen Laden des Programms unbedingt die oben genannte POKE-Zeile eingeben!

Wenn Sie alles richtig gemacht haben und das Programm fehlerfrei abgetippt wurde, speichert es sich nach dem Starten selbst auf Diskette oder Kassette unter dem Namen »MSE V1.0«. Dieses fertige MSE-Programm laden Sie dann bei Bedarf wie ein normales Basic-Programm und starten es mit »RUN«.

So arbeitet man mit dem MSE

Als erstes möchte der MSE den Namen des zu bearbeitenden Programms wissen. Dieser steht in der ersten Zeile unserer MSE-Listings. Dann müssen Sie die Start- und Endadresse des Programms eingeben. Dies sind die letzten beiden vierstelligen Hexadezimalzahlen in der ersten Zeile unserer Listings.

Wenn Sie ein Programm von Diskette oder Kassette laden wollen, um an einer bestimmten Stelle weiterzutippen oder noch eine Korrektur vorzunehmen, geben Sie auf die Frage nach der Startadresse ein »L« ein. Danach müssen Sie <D> oder <T> drücken, je nachdem, ob Sie von Diskette oder Kassette (»tape«) laden möchten. Wenn das Programm unter diesem Namen nicht auf der Diskette vorhanden ist oder ein sonstiger Ladefehler vorlag, meldet sich der MSE mit »I/O-ERROR«. In diesem Fall drücken Sie <RUN/STOP RESTORE> und geben einfach noch einmal »RUN« ein.

Beim Abtippen geben Sie nach und nach die abgedruckten Buchstaben und Zahlen des jeweiligen Listings ohne die Freiräume dazwischen ein. Wenn Sie in einer Zeile einen Tippfehler gemacht haben, meldet sich der MSE sofort mit einem Brummton und der Meldung »EINGABEFEHLER«. Nach einem Druck auf die RETURN-Taste können Sie mit der DEL-Taste den Fehler korrigieren. Wenn Sie das gewünschte Programm vollständig eingegeben haben, speichert es der MSE automatisch auf Diskette oder Kassette.

Bei längeren Listings ist es unwahrscheinlich, daß Sie das komplette Programm auf einmal eingeben. Sie können Ihre bisherige Tipparbeit jederzeit durch <CTRL S> auf Diskette oder Kassette speichern und Ihr Werk später fortsetzen. Sie sollten sich dann allerdings im Heft markieren, wie weit Sie beim Abtippen gekommen sind! Später geben Sie dann nach dem Laden des ersten Programmteils <CTRL N> ein und auf die dann folgende Frage nach der Startadresse die Zeilennummer (Adresse), bei der Sie aufgehört haben zu tippen.

<CTRL M> erlaubt Ihnen jederzeit, Ihr Werk listen zu lassen. Durch <SPACE> können Sie weiterlisten lassen und durch <RUN/STOP> das Listen abbrechen.

Wenn Sie einen Drucker besitzen, können Sie das Programm auch mit <CTRL P> ausdrucken. Mit <CTRL L> wird das Programm noch einmal neu in Ihren C 64 geladen.

(F. Lonczewski/N. Mann/D. Weineck/tr)

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Geschäftsführender Chefredakteur: Michael Scharfenberger

Chefredakteur: Albert Absmeier
Stellv. Chefredakteur: Georg Klinge
Leitender Redakteur: Gottfried Knechtel (kn)
Redaktion: Klaus Schrödl (sk)

Hotline: Gerd Donaubauer, Monika Welzel (640)

Mitarbeiter der Redaktion: Andreas Lietz, Michael Thomas, Dr. Rudolf Egg, Nikolaus Heusler, Ulrich Beenen, Thomas Lipp, Steffan Willmeroth

Redaktionsassistenz: Andrea Kaltenhauser (202)

Layout: Leo Eder (Leitung), Rolf Raß (Cheflayouter)
Andrea Miller, Katja Milles

Fotografie: Jens Jancke

Titelgestaltung: Andrea Miller, Norbert Raab (Air-Brush)

Produktionsleiter: Klaus Buck

Anzeigenverkaufsleitung: Ralph-Peter Rauchfuss

Anzeigenverkauf: Britta Fiebig (282)

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG,
Kollerstr. 3, CH-6300 Zug,
Tel. 042-41 56 56, Telex: 862329
USA: M&T Publishing Inc.; 501 Galveston Drive Redwood City,
CA 94063
Telefon: (415) 366-3600

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen und dazu, daß Markt&Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Marketingleiter: Hans Hörl (114)

Vertriebsleiter: Helmut Grünfeldt (189)

Anzeigenverwaltung und Disposition: Lisa Landthaler (233)

Druck: SOV St. Otto-Verlag GmbH,
Laubanger 23, 8600 Bamberg

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Preis: Das Einzelheft kostet DM 14,–

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1, Telefon (0711) 6483-0

Verantwortlich:
Für redaktionellen Teil: Albert Absmeier
Für Anzeigen: Britta Fiebig

Redaktionsdirektor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft,
Hans-Pinsel-Straße 2, 8013 Haar bei München,
Telefon (089) 4613-0, Telex 5-22052

ISSN 0931-8933

EINSTECKEN UND VOLL LOSLEGEN. AMIGA 500 ORIGINALZUBEHÖR.

COMMODORE AMIGA FEVER STECKT ALLE AN

Farbmonitor A 1084
Das Schärfste für Ihren Amiga.
2.000 Zeichen. 14" Bildschirm. Entspiegelt.

Farbdrucker MPS 1500 C
Da geht's bunt zu. Auf Wunsch auch schwarz.
120 Zeichen/Sek. Normal- und Schönschrift.

A 520 Video-Adapter
Für alle, die TV oder Videomonitor an den A 500 anschließen wollen.

COMPUTER DES JAHRES '87 AMIGA 500 Commodore

Speichererweiterung A 501. Heavy tuning mit clockset! Erweitert RAM um 512 KB auf 1 MB.

A 1010 Disk Drive
Damit geht's rund. Mit 3,5",
Speicherkapazität 880 KB.

Amiga 500
Klasse Software – Riesenauswahl!
Von wem und wie?
Steht alles im Amiga Katalog. Sofort beim Händler besorgen!

Wenn Sie ohne Fehlstarts sofort loslegen wollen, verwenden Sie nur Original-Peripherie von Commodore. Damit läuft Ihr Amiga 500 zur Höchstform auf. Einstecken – und schon geht die Post ab. Denn da paßt jedes Teil zum anderen.
Commodore – Marktführer bei Mikrocomputern.



COMMODORE AMIGA 500 – FÜR ALLE, DIE IMMER VORNE SIND

